

MacTech Magazine
Vol. 17, No. 02 • February 2001

MacTech[®]

The Journal of Macintosh Technology and Development

Safe Haquery, Categorically Speaking

The art of clever programming

by Andrew C. Stone



\$8.95 US
\$12.95 Canada
ISSN 1067-8360
Printed in U.S.A.



LA to New York

(and Seattle and Miami)

in 150 milliseconds or less*



Call for a TeraMedia® demo

TeraGlobal
COMMUNICATIONS

(858) 404.5500 ext 5660
www.teraglobal.com
sales@teraglobal.com



Interact. Collaborate. It's highest-quality multi-point video conferencing and true real-time multimedia collaboration. It's all done in software. Any number of users. Over any distance. At any time. In real time. It flies, you don't.

* Actual video latency varies with bandwidth. In most cases, TeraMedia® latency is below the threshold of human perception.

TeraMedia
by TeraGlobal

*"Never forget that the most powerful
force on earth is love."*

—Nelson A. Rockefeller

Move over love.



Get a new Apple PowerMac™ G4
or one of our refurbished models
at price you will L-O-V-E.



**Small Dog
Electronics**

1673 Main Street
Waitsfield, VT 05673 USA
Phone: **802-496-7171**
Online: **smalldog.com**



Apple Specialist

How To Communicate With Us

In this electronic age, the art of communication has become both easier and more complicated. Is it any surprise that we prefer **e-mail**?

If you have any questions, feel free to call us at 805/494-9797 or fax us at 805/494-9798.

If you would like a subscription or need customer service, feel free to contact Developer Depot Customer Service at 877-MACTECH

DEPARTMENTS

Orders, Circulation, & Customer Service

Press Releases

Ad Sales

Editorial

Programmer's Challenge

Online Support

Accounting

Marketing

General

Web Site (articles, info, URLs and more...)

E-Mail/URL

cust_service@mactech.com

press_releases@mactech.com

ad_sales@mactech.com

editorial@mactech.com

prog_challenge@mactech.com

online@mactech.com

accounting@mactech.com

marketing@mactech.com

info@mactech.com

http://www.mactech.com

The MacTech Editorial Staff

Publisher • Neil Ticktin

Managing Editor • Jessica Stubblefield

Online Editor • Jeff Clites

Regular Columnists

Getting Started – Networking

by John C. Welch

Programmer's Challenge

by Bob Boonstra

MacTech Online

by Jeff Clites

From the Factory Floor

by Metrowerks

Regular Contributors

Vicki Brown, Andrew Stone, Tim Monroe, Erick Tejkowski, Kas Thomas, Will Porter, Paul E. Sevinç, Tom Djajadiningrat, and Jordan Dea-Mattson

MacTech's Board of Advisors

Dave Mark, Jordan Dea-Mattson, Jim Straus, Jon Wiederspan, and Eric Gundrum

MacTech's Contributing Editors

- Jim Black
- Michael Brian Bentley
- Tantek Çelik, Microsoft Corporation
- Marshall Clow
- John. C. Daub
- Tom Djajadiningrat
- Bill Doerrfeld, Blueworld
- Andrew S. Downs
- Richard V. Ford, Packeteer
- Gordon Garb, Cobalt Networks
- John Hanay, Apple Computer
- Lorca Hanns, San Francisco State University
- Ilene Hoffman
- Chris Kilbourn, Digital Forest
- Scott Knaster, Microsoft
- Mark Kriegsmann, Clearway Technologies
- Peter N. Lewis, Stairways Software
- Bill McGlasson, Apple Computer
- Rich Morin
- Terje Norderhaug, Media*Design-In-Progress
- Nathan Nunn, Purity Software
- John O'Fallon, Maxum Development
- Alan Oppenheimer, Open Door Networks
- Avi Rappoport, Search Tools Consulting
- Ty Shipman, Kagi
- Chuck Shotton, BIAP Systems
- Cal Simone, Main Event Software
- Steve Sisak, Codewell Corporation
- Dori Smith
- Andrew C. Stone, www.stone.com
- Michael Swan, Neon Software
- Chuck Von Rospach, Plaidworks
- Bill von Hagen
- Eric Zelenka

Xplain Corporation Staff

Chief Executive Officer • Neil Ticktin

President • Andrea J. Sniderman

Controller • Michael Friedman

Production Manager • Jessica Stubblefield

Production/Layout • W2 Graphics

Marketing Managers

Nick DeMello and Alyse Yourist

Events Manager • Susan M. Worley

Network Administrator • Ben Baumer

Accounting • Jan Webber, Marcie Moriarty

Customer Relations • Laura Lane

Susan Pomrantz

Shipping/Receiving • Joel Licardie

Board of Advisors • Steven Geller, Blake Park, and Alan Carsrud



This publication is printed on paper with recycled content.

All contents are Copyright 1984-2001 by Xplain Corporation. All rights reserved. MacTech and Developer Depot are registered trademarks of Xplain Corporation. RadGad, Useful Gifts and Gadgets, Xplain, Depot, The Depot, Depot Store, Video Depot, Movie Depot, Palm Depot, Game Depot, Flashlight Depot, Explain It, MacDev-1, THINK Reference, NetProfessional, NetProLive, JavaTech, WebTech, BeTech, LinuxTech, MacTech Central and the MacTutorMan are trademarks or service marks of Xplain Corporation. Sprocket is a registered trademark of eSprocket Corporation. Other trademarks and copyrights appearing in this printing or software remain the property of their respective holders.

MacTech Magazine (ISSN: 1067-8360 / USPS: 010-227) is published monthly by Xplain Corporation, 850-P Hampshire Road, Westlake Village, CA 91361-2800. Voice: 805/494-9797, FAX: 805/494-9798. Domestic subscription rates are \$47.00 per year. Canadian subscriptions are \$59.00 per year. All other international subscriptions are \$97.00 per year. Domestic source code disk subscriptions are \$77 per year. All international disk subscriptions are \$97.00 a year. Please remit in U.S. funds only. Periodical postage is paid at Thousand Oaks, CA and at additional mailing office.

POSTMASTER: Send address changes to **MacTech Magazine**, P.O. Box 5200, Westlake Village, CA 91359-5200.

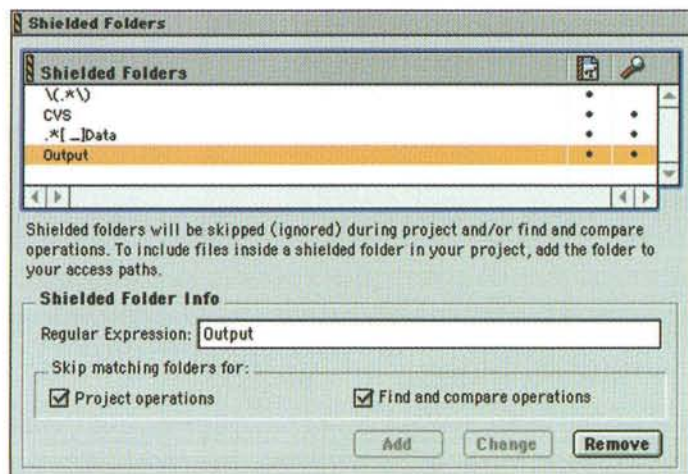
C o n t e n t s

February 2001 • Volume 17, Issue 02

Feature Articles

25 **ADC DIRECT** Tools for the Trade

30 **QUICKTIME TOOLKIT** Honey, I Shrunk the Kids by Tim Monroe



From the Factory FloorPage 10

C o l u m n s

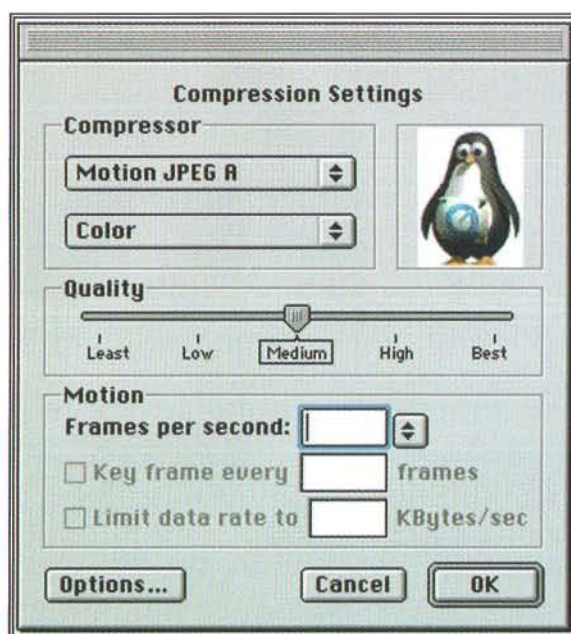
VIEWPOINT

4 **Macworld Expo SF 2001** by John C. Welch

10 **FROM THE FACTORY FLOOR** CodeWarrior Version 6.0 by Richard Atwell

54 **PROGRAMMER'S CHALLENGE** Trilite by Bob Boonstra

76 **MACTECH ONLINE** The New Basics: WebDAV by Jeff Clites



QuickTime ToolkitPage 30

NETWORK MANAGEMENT

16 **Networking the OS X** Public Beta by John C. Welch

COVER STORY

70 **The Art of Safe Haquery,** Categorically Speaking by Andrew Stone



The Art of Safe Haquery.....Page 70

By John C. Welch

MacWorld Expo S.F. 2001

Geeks In Toyland

WELCOME

Well, MacWorld Expo in San Francisco is the traditional way for Mac users of all stripes to start the year off right, and this year's was exceptionally good. From the plethora of OSX announcements to the amazing attendance figures, (over 93,000, a record for any MacWorld anywhere), this was a show to remember. There were hardware and software announcements galore, and products to meet any Mac user's needs, including those of the network administrator. So let's start where any MacWorld starts; the Keynote.

OPENING KEYNOTE

A Steve Jobs keynote is something that we all look forward to, and this one was easily the equal of any expectations. Although the entire computer industry was eviscerated by one of the worst holiday quarters on record, as always, bad news from Apple is looked at with particular emphasis. Ignoring the industry downturns, and the first bad quarter in years, the familiar 'Will Apple survive' refrain was starting to show up again, requiring an exceptional keynote from Jobs to combat it, and that is what we got.

First of all, there was the contriteness regarding missing the boat on CD-R/W drives. No excuses, just an admission that Apple had messed up, but that they were going to fix that. Dovetailing in with the announcement that CD-R drives were going to be standard on almost all model of G4 tower was the announcements of the new G4 towers themselves. The new towers have CPU speeds of 466MHz, 533MHz, 667MHz, and 733MHz. So now, the slowest G4 will be almost as fast as the previous high end model. But as we will see, other non - CPU related improvements may actually make this model faster than the 500MHz previous high end. However, the changes to the CPUs in the towers are significant by themselves.

New G4s

The two lower end models of G4 tower appear to be using the MPC7410 from Motorola. The 7410 is almost identical to the previously used 7400, but with changes designed to allow it to be used in low-heat/embedded situations. The 7410 drops the 7400's

support for 3.3V interface voltage, for example. The 7410 is built on a .18µm CMOS process, an improvement on the .20µm that the 7400 is built on. The die size on the 7410 is smaller, 52mm² down from the 83mm² size of the 7410. The result of this is a chip that is capable of operating cooler than the 7400. But in most other ways, the 7410 is simply a 7400 that needs less power and emits less heat.

The two high end G4 towers are using the MPC7450, also from Motorola, and this is the chip that Mac users have been waiting quite some time for. Improvements to the 7450 include a pipeline increase from four to seven stages, allowing for the higher clock speeds, the addition of four new execution units, support for 4 instructions per clock cycle, up from the 7410's three, on-die 256K L2 cache, increasing the L1 cache bus from 128 to 256 bits, support for a faster system bus, and support for up to 2MB of L3 cache. The 7450 sports two more AltiVec units, making the total 4, and can fill two of them every clock cycle, again, an improvement from previous G4 implementations. (This is where things get tricky. For example, I can fill 256 bits worth of AltiVec unit every clock cycle. Each of those 128 bit instructions can be four 32 bit integer values in a packed data type. So, in a sense, I can actually run 8 integer ops per clock cycle. There's more to it than that, but things like this really make MHz comparisons quite simplistic.) In addition to the new AltiVec units the number of integer units has gone from 2 to 4, and this will give almost all applications better performance, not just the AltiVec-enabled ones. In short, the 7450 is a much more powerful chip than its predecessors, and with the new design, and an improved process, hopefully, we'll be seeing regular speed improvements from now on.

But there is more than just the CPU improvements to the new G4s. The system bus speed has been improved by a third, up to 133MHz from the previous 100MHz. The PCI subsystem, in addition to gaining a slot, for a total of four, also gets PCI Write Combining. This allows sequential write transactions, either Memory Write, or Memory Write and Invalidate commands, to be combined into a single PCI transaction. This give the G4 towers some long awaited PCI performance gains. The AGP bus goes from 2X to 4X, and we get nVidia GeForce 2 MX cards on all but the lowest end G4s, which will use the ATI Radeons. Finally, for those who need or want, the 533MHz model comes in an optional dual CPU multiprocessor

John Welch <jwelch@aer.com> is the Mac and PC Administrator for AER Inc., a weather and atmospheric science company in Cambridge, Mass. He has over fifteen years of experience at making computers work. His specialties are figuring out ways to make the Mac do what nobody thinks it can, and showing that the Mac is the superior administrative platform.



Control your web database development destiny. today.

Today is the day you take control of your destiny
as a professional application developer.
No excuses. You're going to do it today, **Period.**

Your new destiny begins with a look at the development tools you're using. Are they open? flexible? scalable? Do you always worry that your tools can't accomplish the job? If your tools limit your ability to deliver solutions, how can you realize your true potential?

The tool to help you get the most out of your potential is 4th Dimension. The 4th Dimension (4D) product line is a carefully integrated, yet open development environment that offers unprecedented flexibility and scalability. With the release of 4D version 6.7 there's never a need to worry about the ability to accomplish the job at hand. 4D supports a massive range of solutions including web access, SSL, XML, WAP, FTP, Email and more. When your client asks for connectivity to other RDBMS systems like ORACLE™ or

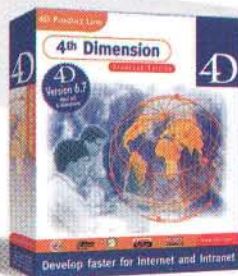
MS SQL Server™ you simply won't have to worry - 4D has been providing connectivity for over a decade. What if you could offer that same client security for the data in the database in the event the database was stolen? With 4D Version 6.7 we've pioneered the use of SSL-based technology to shore-up and encrypt the data in the database. Yet there's so much more.

Are you ready to take control, to deliver versatile solutions that meet or exceed your client expectations? You can take that first step by downloading the demo of 4D Version 6.7. The demo includes a tutorial that quickly immerses you in 4D and targets you in the right direction.

Here's to your new destiny using 4D.

www.4D.com/destiny

1.800.881.3466 opt 3



4th Dimension v6.7

RAD • RDBMS • CGI • HTTP • SSL

model. According to Jobs, had Apple tried to initially offer the higher end models in MP configurations, he would have been introducing them in April. This candid admission that supplies of the new toys are going to be strained at first is unusual, but refreshing, and shows that Apple is trying to sooth its customers, and by extension, its investors that while the news this quarter may not be good, that's not a permanent feature.

Titanium

Well, it's not often you hear the CEO of a major computer company talking about how they want sex as well as power in a new product. It's even rarer to actually see them pull it off. But the PowerBook G4 is just that. Power is the MPC7410 that is running the laptop, and sex is everything else. What's even more amazing is how many segments of the laptop market this product is suited for. You're a CEO who wants something stylish, yet thin enough to fit in that oh-so stylish case? Titanium. You're a traveler who wants something light, with a big screen, yet not so tall that you can't use it on a plane? Titanium. You're a mobile media type who needs FireWire, USB, and a decent aspect ratio on your screen to see your work? Titanium. Really, about the only segments this doesn't appeal to are those who desperately need internal bays, and those who want a laptop not much bigger than a Newton. (Speaking as someone with a plethora of bay devices, I'll trade them in for what I would get out of a Titanium, in a heartbeat.) Judging by the frenzied cell phone orders I heard leaving the keynote, I think this will be a winner for Apple.

OS X

Of course, you can't have a MacWorld keynote without a demonstration of the latest versions of OS X. This time was no different. Steve was happily showing off the latest features of the OS, mostly those that had received the most number of comments. The Apple Menu, while not the same Apple Menu from the current Mac OS, is back where we are used to it, and it now has some spiffy new features that are available at all times, regardless of application, such as the Special Menu functions, Location Manager, etc. The Dock has been modified to allow for hierarchal popup menus from folders in the Dock, giving back a couple of missed features missing in the PB. Ctrl-clicking different items in the Dock now works as well, for things such as monitor resolution. So without simply jamming the current Apple Menu and Control Strip in the OS, Apple is still managing to restore much of that functionality in a coherent and integrated way. The speed of OS X seems to have been improved, and reports from other sources indicate that a lot of optimization work is definitely paying off.

In the end, it was a successful keynote, both for product and image. Steve showed that Apple is not just sitting on its keister, and is actively working to answer the criticism its been facing lately.

MICROSOFT KEYNOTE

The second day of the show started with an unusual event: a non-Apple keynote. This was Microsoft's keynote, and although it was held in the Marriott instead of the Moscone, it was well attended and well received. There were two reasons for this. The first being the public unveiling of Outlook 2001, the Mac version of the

Exchange client. Looking exactly like its Windows cousin, for once, Mac users on an Exchange server won't be feeling like a bad smell under the couch. Finally the calendaring and scheduling functions on the Mac are feature as feature complete as the Windows version, to the point where a Mac client can even alter a Windows client's calendar. The Mac version sports some UI improvements over the Windows version, most notably in simplifying the sometimes mind-numbing number of dialogues that some tasks can cause. There is also a new wizard that runs when you first install the application, that greatly simplifies the process of connecting to your Exchange server. With the short shrift that Mac users sometimes get from IS in the enterprise, Microsoft was smart to take the lead on this. Finally, in response to my query about how will the Mac version deal with Melissa, and other VBA - created virii, the response from Microsoft, was, "Easy, we don't support Visual Basic in the product, but we are going to try and have an excellent AppleScript implementation." They are quite aware of the AppleScript standard set by such products as Outlook Express and Entourage, and are using them as an idea of where Outlook 2001 should be.

The other half of the keynote was devoted to Office 2001, and specifically, demonstrating Office running as a Carbon application. Although obviously an early alpha, it was Office, it was Carbonized, and it was running. Evidently stung by recent criticism of Microsoft's commitment to the Mac OS, not only did we see the Office on X demo, but also some slides showing just how large an undertaking that Carbonizing Office is. Although, not surprisingly, no hard release date was given, Microsoft is saying Fall of 2001 as a release timeframe.

Like the Apple keynote from the day before, the Microsoft keynote managed to not only introduce new product, but also answer criticism, and show that Microsoft's commitment to the Mac OS continues unabated.

OTHER NON-ANNOUNCEMENTS

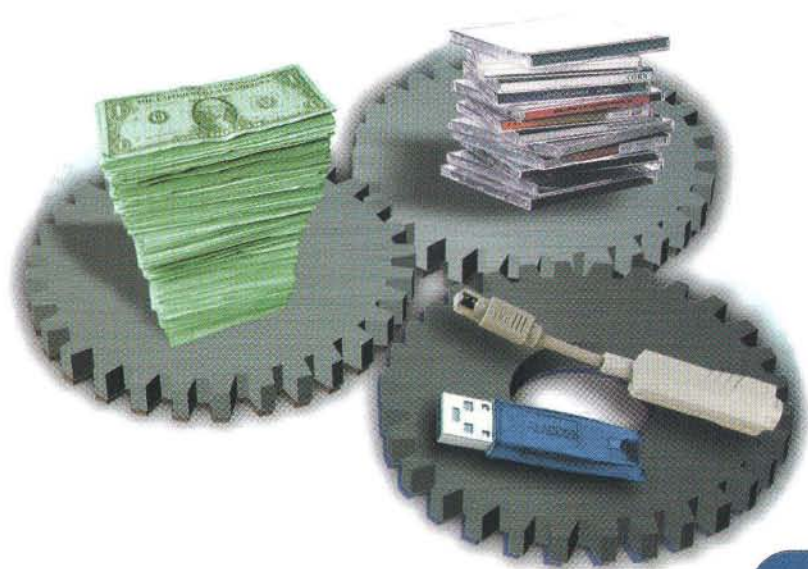
Quiet release and MacWorld Expo go together like Sodium and Water, but this show had at least two rather major products that just sort of showed up, with almost no fanfare. This first of these was Mac OS 9.1, the latest version of the current Mac OS. Although not a radical change, like OS 9.0 was, there are some rather major changes in this release.

The Finder as a number of new features, such as:

- Not protesting bad memory settings in the Get Info window until you close it,
- A context menu that allows you to open OS X packages as folders
- A Window menu, that shows current Finder windows.
- The Desktop folder can now be redirected to volumes other than the startup volume.
- It is also now not possible to copy an alias file over a document file.
- Share point names can be no longer than 27 characters
- If you edit a file's name in the Get Info window and the lock it, the name change sticks
- The 'Encrypt' command no longer works on aliases

Software Developers

HASP4 for Mac - Turn Your Lost Software Into Profits!



**NOW
SUPPORTS
OS X!**

Turn your lost software into profits. By preventing unauthorized software usage and duplication, HASP4 for Mac protects intellectual property rights and increases revenues. It enforces the legitimate use and licensing of software applications. HASP4 for Mac controls software distribution and expands sales channels. Companies such as 3M, Quark, Lucent, HP, Cadlink and Scitex know that they can count on HASP to protect their software products.

Order your HASP4 for Mac Developer's Kit at:

www.eAladdin.com/mactech

**Or call 800-223-4277 EST or
800-562-2543 WST, CST & MST.**

ALADDIN®

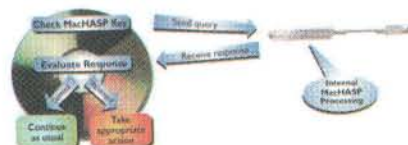
Securing the Global Village

Identification Card

HASP4
The Professional Software
Protection System

Name:	HASP4 for Mac
Main Mission:	To protect your software & Increase your revenues
HASP is:	A small hardware key that connects to the computer and prevents unauthorized use of your software.
Features & benefits	Prevents software piracy Controls software distribution Protects intellectual property Increases revenues Reduces software distribution costs

Product Diagram:



International HASP@eAladdin.com North America HASPus@eAladdin.com UK HASPuk@eAladdin.com Germany HASPde@eAladdin.com
Benelux HASPnl@eAladdin.com France HASPfr@eAladdin.com Israel HASPil@eAladdin.com South Africa HASPza@eAladdin.com
Brazil HASPbr@eAladdin.com Japan HASPjp@eAladdin.com Russia HASPru@eAladdin.com Spain HASPes@eAladdin.com China HASPcn@eAladdin.com

- You can't bless the Desktop folder by moving the System and Finder there
- More intelligent identification of OS X packages
- Custom Icon support for OS X packages
- Apple Events issues with background windows have been fixed

There are a lot more fixes and features across the entire OS, and those are detailed in Technical Note 2010, available on Apple's developer site.

The second non-announcement was the showing of the Aqua-ized release of OS X Server. The most obvious change is the integration of OS X Server with Aqua, and the more modern underpinnings of OS X. But along with that come some features that network administrators will love just as much:

- Native support for SMB/CIFS Windows networking
- Support for SMB printing
- Improved UI for management tasks
- Better SLP support
- OS X support with NetBoot and Macintosh Manager
- Integrated LDAP support
- A POP/IMAP mail server
- Improved Apache support, such as WebDAV

This is also looking to be the version of OS X Server that acts as the integration between earlier versions of OS X Server and AppleShareIP, which will simplify the task of figuring out which server product to use. If the integration is done as well as Apple says it will be done, then this will be A Good Thing, but only time will tell.

FLOOR PRODUCTS

This MacWorld was centered around one thing...OS X. Almost every product of interest to IS or corporate computing was showing off an OS X version, or an early beta of an OS X version, or something having to do with OS X. In fact, there were so many, I couldn't even come close to seeing them all. Among these were products I use as well, such as WebSTAR, FileMaker Pro, and other products, like 4D, Rewind, InterMapper, and QuickKeys.

One of the more heavily hyped products was AccountEdge, by MYOB. This product is MYOB's attempt to not only take care of existing QuickBooks customers abandoned by Intuit, but to bring back some of the small businesses that may have left the Mac because of QuickBooks. Currently available as a Classic Mac OS application, there is a Carbon version being developed, so that users of that OS will have an accounting solution as a native application. AccountEdge also features improved integration with Microsoft Office 2001, and looks to be a winner of a product.

Alsoft was selling the latest update to DiskWarrior, version 2.1, which, although not yet Carbonized, is able to fully repair a Mac OS X HFS+ disk. It also features quite a few upgrades for changes in Mac OS 9.1 as well. The directory

comparison feature has been improved, and the report format changed to be more readable. Shortly after the Expo, Alsoft announced that the long awaited update to Disk Express Pro will be released as a native OS X and Classic Mac OS application, and will be free to all customers who owned DiskExpress Pro 3 when Mac OS 8.1 was released.

In addition to Alsoft, MicroMat was demonstrating Drive 10, a Mac OS X native application, that not only checks and repairs volume structures, but also runs other drive hardware diagnostics, and is able to interface with the SMART circuitry on those drives that have it. This allows Drive 10 to monitor issues like the quality of power being delivered to the drive, and other low level problems that can often appear as other errors. Having personally been bitten by a bad power supply, that Drive 10 could have possibly caught, I plan on getting a copy as soon as it is released.


Dantz was showing off the Carbonized version of Retrospect 4.3, along with the OS X - native backup agent. Both run natively under OS X, but the agent is a Cocoa application, and interestingly enough, by using Cocoa, Dantz says they get a 5x speed improvement over both the Windows and Mac OS agents. The first release of Retrospect will be a Carbon version of the current Mac OS server, so that network administrators have a native backup solution as quickly as possible. The next iteration of the server will take the lessons learned in creating the Windows server, and apply them to OS X's superior GUI and feature set, allowing it to leapfrog the Windows server.

Netopia, recently bought by Proxim, was showing the OS X versions of both Timbuktu Pro and netOctopus. Both are being pushed to OS X as fast as possible, although due to the nature of both of these applications, final development has to wait for the GM release of OS X. At the netOctopus User's Luncheon, Netopia presented a very detailed roadmap for netOctopus's future development, talking about the new features being planned for the product, and the integration with other network management products and systems.

Late Night Software was showing off the OS X native version of its excellent AppleScript IDE, Script Debugger, a major product for those of us who use AppleScript on a daily basis.

Finally, WildPackets, formerly the AG Group, was demonstrating the OS X native version of EtherPeek, which, like many of the products I have talked about, may not be sexy, but is absolutely critical to most network managers.

CONCLUSIONS

For network managers, this was a major show indeed. We finally get to actually see the products we care about running in OS X, instead of the usual "Oh we'll have it ready real soon" promises. There is something reassuring about being able to see, and in most cases, actually use the products that we need to move to OS X. If this show was any indication, MacWorld in New York is going to be a network manager's toy store. 

Looking for more speed, scalability and reliability from your database? **c-tree Plus**® is the perfect fit!

What shape is your database project? A tiny embedded database that requires a small footprint but demands rigorous functionality? Or a huge multi-platform, multi-user, multi-headache project? Have you struggled to find a better solution for your data access requirements, only to feel like you're sacrificing one requirement over another?

FairCom has delivered uncompromising database technology to commercial developers for over twenty years, with the primary goal of keeping control in the hands of the developer. Our customers rely on the speed, flexibility, scalability and reliability of c-tree Plus. The high performance and low cost of ownership make c-tree Plus an excellent choice for all sizes of database development projects.

Proven Database Technology

Besides the ISAM-level control and speed, perhaps the most important reason why small development houses and Fortune 1000 companies have chosen FairCom technology is the superior service offered by our sales and support staff. Our development staff understands the challenges you face, and we'll put our twenty years of experience to work to help you build better solutions.

Single Solution for Diverse Implementation

- Vertical Markets
- Embedded Systems
- Web & ASP Markets
- E-Commerce
- Smart-Cards
- Web-Enabled Appliances

Comprehensive Feature Set

- Royalty-free single user and multi-user support with full source code
- Client/server and custom server support
- Robust server side SDK - build your own application specific database server with full source available!
- Thread compliant + portable thread API
- Full featured transaction processing with savepoints, abort and full rollback
- Comprehensive security and encryption features
- Small footprint
- Fixed and variable length records and keys
- Store any data type - up to 18 million terabytes!
- Dynamic space reclamation
- ODBC and Crystal Reports™ drivers
- Full ISAM functionality
- Powerful stand-alone index support

Multiple-Platform Support

Mac OS, Mac OSX, MkLinux, Linux (PPC, Intel, SPARC, Alpha), Windows 95/98/ME/2000/NT, Novell Netware, Solaris (SPARC, Intel), Sun OS, OS/2, AIX, HP UX, SCO UnixWare, Interactive, AT&T Sys V, QNX, 88OPEN, FreeBSD, Lynx, Banyan Vines, and more...

Supports ADSP, SPX, TCP/IP



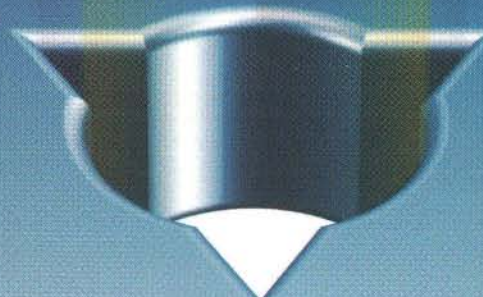
FairCom®

QUALITY • THROUGH • EXPERIENCE

www.faircom.com/maca

FairCom Offices

USA	573.445.6833
EUROPE	+39.035.773.464
JAPAN	+81.59.229.7504
BRAZIL	+55.11.3872.9802



© 2000 FairCom Corporation. Other company and product names are registered trademarks or trademarks of their respective owners.

Mac Support Since 1985 • USA. 800.234.8180 • info@faircom.com

By Richard Atwell, ©2001 by Metrowerks, Inc., all rights reserved

CodeWarrior Version 6.0

It's been a while since I last wrote an article for this column. Around the time of my last article CodeWarrior Version 6.0 was in the final stages of development and we decided to devote all of our energies to making what we think is the strongest release ever. For the next while this column is going to be every other month.

In this column I'd like to catch up on the latest release.

IN THE BEGINNING

CodeWarrior has been the choice of many Macintosh developers since late 1993. Since that time there have been 17 major releases: three DR, eight Gold and six Professional. Although we've changed the name along the way, the goal has remained the same: provide best of breed tools for the serious Macintosh application developer.

This time around, we felt so confident about the latest release that we decided to use it for its own development. That is, with the exception of the compilers themselves, everything on the Pro 6 CDs was developed and built internally with Version 6.0.

If you're new to CodeWarrior, Metrowerks offers seamless integration of standards-compliant compilers, clever linkers, a powerful editor, project manager, visual interface constructor, modern application framework, profilers and debuggers in a single package.

VERSION 6.0

With Version 6.0, our focus has been Carbon compliance and initial support for the as yet unfinished Mac OS X. Using the same Universal Interfaces that we shipped on the CDs, we carbonized the IDE and all the accompanying tools for Mac OS X Public Beta. This Carbonization isn't 100% complete, but we'll catch up before Mac OS X goes final.

CodeWarrior now requires Mac OS 8.6 or higher because of its Carbonization. If you remember, Pro 5 was the last release that let you host the IDE on 68K-based Macintosh computers. Given the upcoming trends from Apple and feedback from our customers, we made the decision to make Version 6.0 the last release to support 68K code generation. Going forward, we will support code generation on PowerPC only. We see very few FAT applications being created these days and it has been a number of years since 68K hardware was sold by Apple, so we're going to focus on PowerPC.

NEW FIND DIALOG

Much in the way the Mac OS Finder bears an uncanny resemblance to the original Finder that shipped with the original Mac 128k, the IDE Find dialog seems to have been with us almost unchanged since DR/1. Over the years we've collected plenty of feedback and decided to take the plunge and rewrite it for Version 6.0.

The first thing you'll notice is that we've split the Find Dialog into two separate windows. Accessible from the Search menu, the menu items **Find** and **Replace** and **Find in Files** are available.

Select **Find and Replace** and the window that appears, Figure 1, looks similar to the upper portion of the old Find dialog. This window is for searching editor windows and the window options behave as before with the addition of two new options: **Search Selection Only**, and **Direction**, which lets you control the direction of the search from the insertion point. Gone from the old window is the batch checkbox, which has been replaced by a **Find All** button.

Richard Alexander David Atwell, aka ratwell, is a Mac OS Debugger Engineer at Metrowerks and takes time out from development to keep MacTech readers informed about the world of Metrowerks. Good ideas for CodeWarrior t-shirts can be sent to ratwell@metrowerks.com.

OpenBase SQL 6.5
the high-performance
database for MacOS X

OPENBASE FEATURES

- BC Driver
- WebObjects Adaptors
- ALBasic Plugin
- Multi-Threaded Server
- Low Level Locking
- 100 MB Searchable Blobs
- Database Replication
- Database Management Tools
- Graphical Schema Design Tools
- Remote Administration

process over 2 million
transactions each day.
OpenBase performance has
an outstanding."
FlightArrivals.com

AMAZING SPEED

AMAZING POWER

[AMAZINGLY AFFORDABLE]

With over **2 Million Transactions** per day, **FlightArrivals.com** flies with **OpenBase SQL**

Take OpenBase SQL for a Test Flight Today!
Get Your FREE Developer's License at www.openbase.com

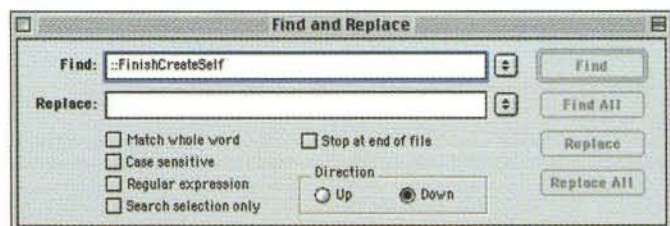


Figure 1. Find and Replace window.

The **Find in Files** window is the more interesting of the two new windows. If you don't like the default key binding that we gave this menu option, just change it using the **Commands & Key Bindings...** menu item from the **Edit** menu.

The **Find in Files** window is a tabbed window. The **In Folders** tab allows you to select a starting folder from which to base your searches. You can selectively search sub-folders and filter your searches by file type.

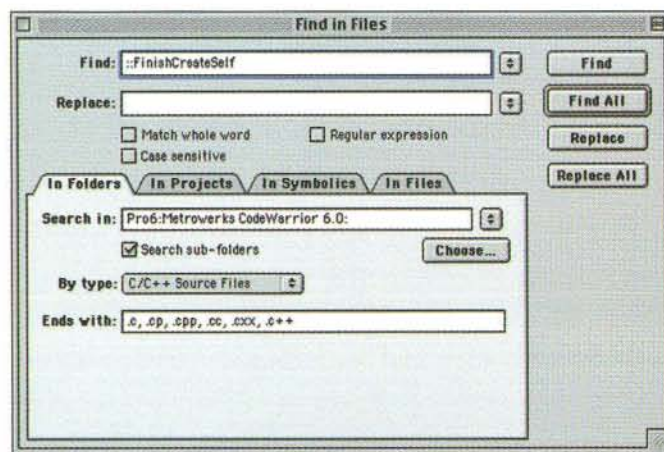


Figure 2. Find In Files window.

The second tab, **In Projects**, is an improvement over the old Find dialog's Project pop-up. It provides the ability to broaden the search to all open projects, or refine the search to specific targets. Also new is the ability to search cached sub-projects. As always, you can filter your search by sources, project headers and system headers.

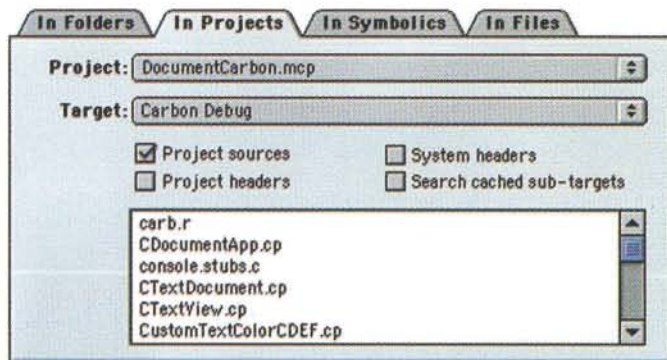


Figure 3. In Projects tab.

In addition, you can now search only files that are in the symbolics for xSYM files that you have open. Searching through your symbolics is efficient, but the list will only contain the files that were actually used to build your targets. Any files in your symbolics that are missing, like sources for libraries built on someone else's machine, are noted in the search results window.

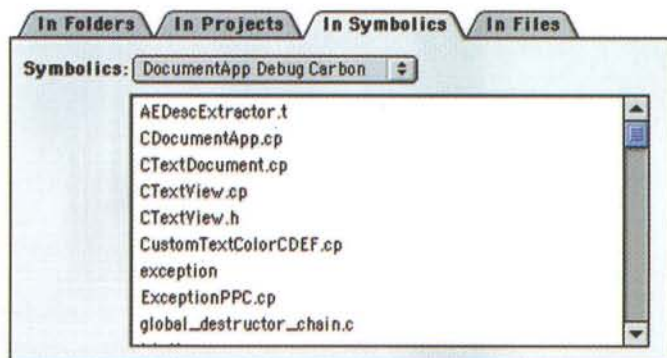


Figure 4. In Symbolics tab.

The last tab is an extension of the old Find dialog's Other button, but you can now search just the open editor files in addition to any set of file you wish to create. You can still drag and drop files and folders into this pane.

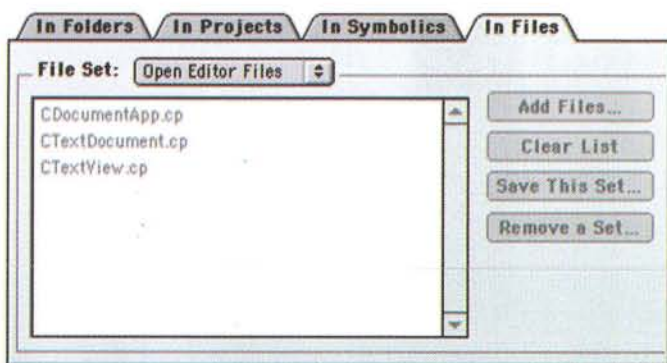


Figure 5. In Files tab.

EDITOR

The editor has been rewritten from the ground up. Although the editor's appearance hasn't really changed, performance has been improved when editing huge files. The insertion point is now tracked by column and a new symbol completion popup is available to help you write code. This is helpful if you've forgotten the names of methods that the browser database collects for your project when you have that build option on.

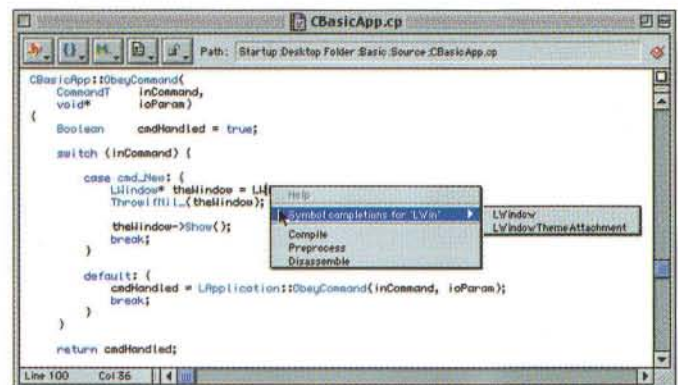


Figure 6. Symbol Completion popup.

NEW TARGET PANELS

A few new panels have been created and improvements have been made to others. A long requested feature has been an option to modify the Finder bits of your target's file name. So, a new target panel was created so you can easily set the shared bit, among other things.

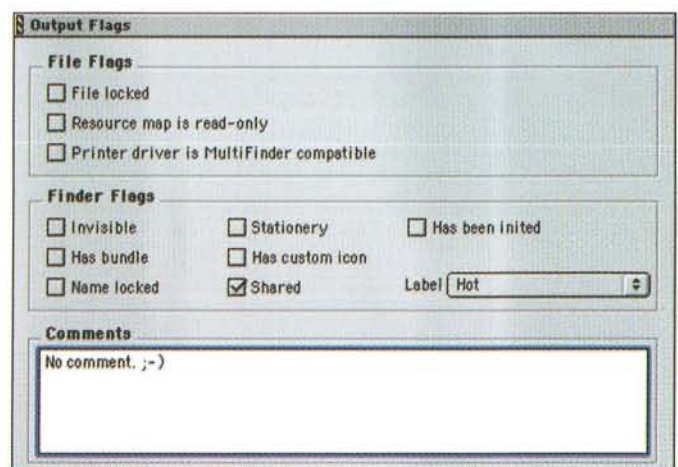


Figure 7. Output Flags target panel.

The other new target panel allow you to create Mac OS 9 packages from your output directory as part of the build process. This capability is provided via post-linker, so be sure to select the Mac OS Package post-linker from your projects target settings panel.

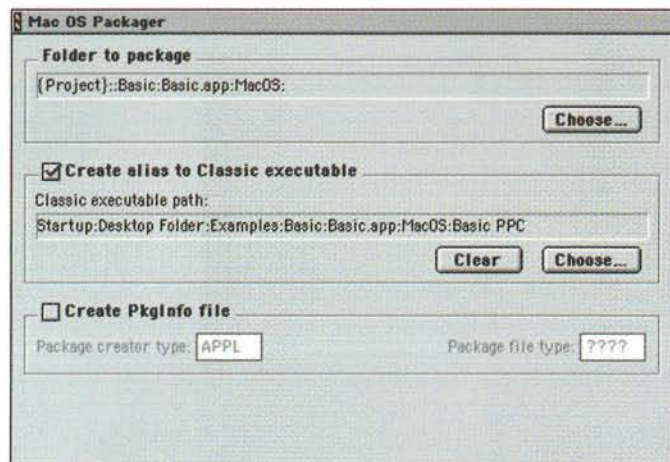


Figure 8. Mac OS Packager target panel.

IMPROVED IDE PREFERENCES

With Version 6.0, we've provided the ability to build projects on locked volumes such as CD-ROMs. You can now find references from the Search menu using Apple's Help Viewer. Select this from the IDE Extras preference panel.

A new panel was created to allow you to further filter your find and compare operations in addition to filtering other project options. One of the default filters we've provided allows you to skip the contents of your invisible CVS folders if you use Mac CVS Pro for version control. The shielded folders panel uses regular expressions to specify your filters so you can easily write a single filter for a variety of project directories that you may want to omit.

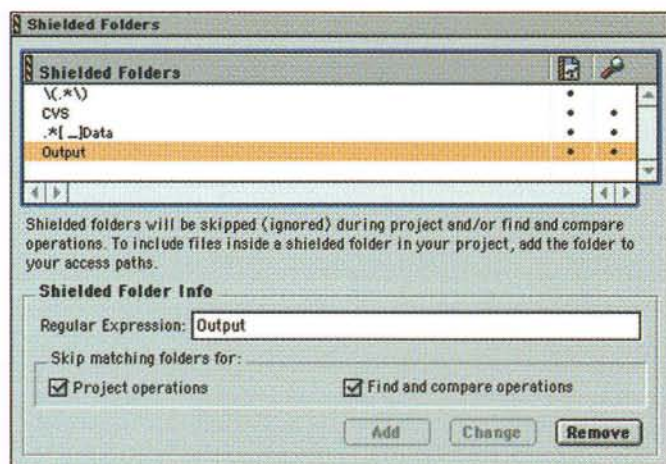
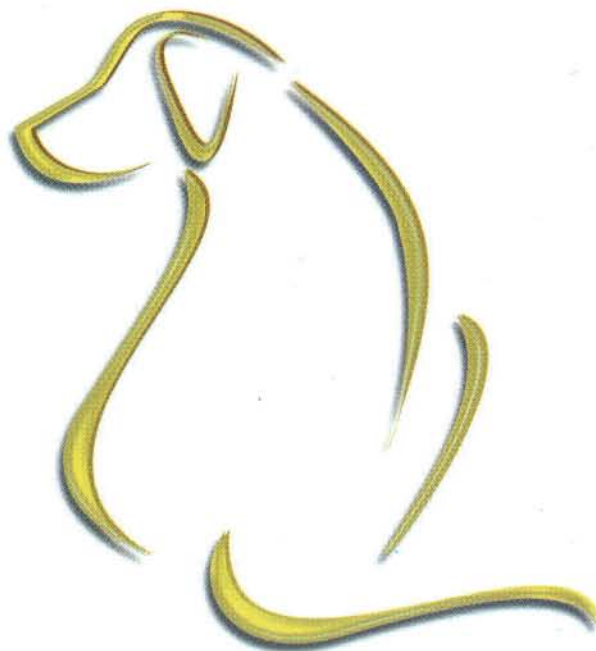


Figure 9. Shielded Folder preference panel.

The last major improvement to the IDE's preference panels is for remote debugging, which we introduced with Pro 5. The IDE 4.0 that shipped with that release forced you to restart the IDE in order to switch between debugging an application on a local machine or a remote machine.

We've removed this limitation with Version 6.0 and to simplify the remote debugger settings we've created an



YDL 2.0
Cuter. Softer. Fluffier.

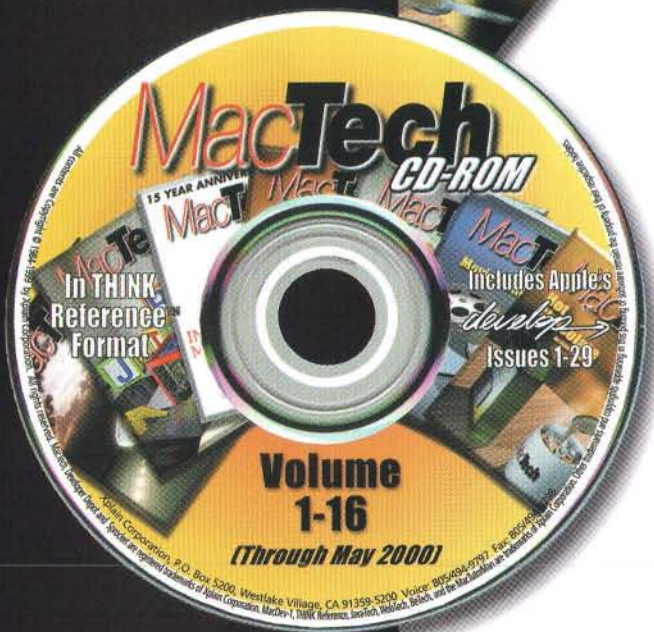
(coming soon)

www.yellowdoglinux.com
by Terra Soft Solutions, Inc.

MacTech CD-ROM

Get the Info you need in one shot!

The MacTech CD-ROM with THINK Reference is the essential reference resource for Macintosh programmers. This CD includes the THINK Reference personal database system and a wealth of Macintosh programming databases, featuring over 160 issues of the journal of Macintosh programming -- MacTech Magazine. This release also features the THINK Reference Compiler, which allows you to compile HTML files into your own compact, searchable THINK Reference databases.



www.mactech.com

PO Box 5200 • Westlake Village, CA • 91359-5200
800/MACDEV-1 (800/622-3381) • Outside US/Canada: 805/494-9797
Fax: 805/494-9798 • E-mail: orders@devdepot.com

address book to store the TCP/IP addresses of the remote machine that you may wish to remote debug.

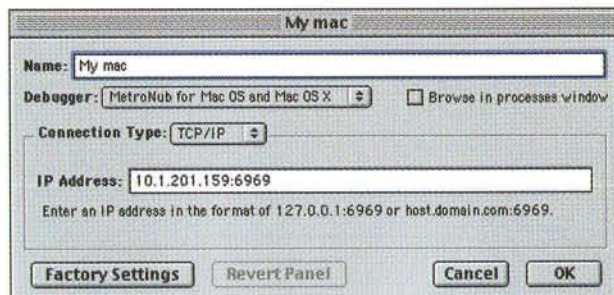


Figure 10. Remote Connections IDE panel.

Once you create a remote connection, you can specify on a per target basis that you wish to remote debug your application. Each target now has a Remote Debugging panel that lets you select the remote connection and specify where you want to transfer the output target on the remote machine and/or the name of an application you wish to launch.

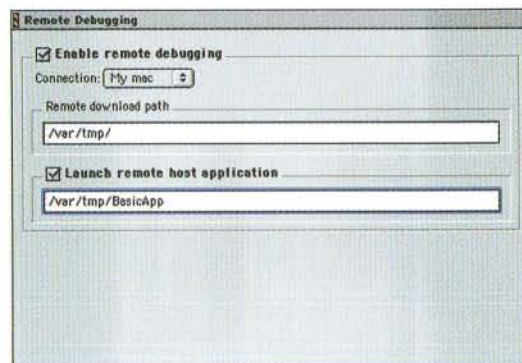


Figure 11. Remote Debugging target panel.

These options make it possible to remote debugging applications on several machines at once in case you ever find the need to debug both client and server applications at the same time. With some additional panel settings you can also download any supporting files such as shared libraries when you start remote debugging.

URLS

- Mac CVS Pro: <http://www.maccvs.org/>

CREDITS

If you'd like to get in touch with us about CodeWarrior issues, post to our newsgroup or email us directly. Visit our website at www.metrowerks.com to learn more about us.

- Newsgroup: comp.sys.mac.programmer.codewarrior
- Technical Support: cw_support@metrowerks.com
- Report Bugs: cw_bug@metrowerks.com
- Suggestions: cw_suggestion@metrowerks.com





Programming Doesn't Have To Be This Difficult. It's REALbasic!

NEW
Version!
REALbasic 2.1*
FREE
Update!

REALbasic is the award-winning, visual, object-oriented BASIC development environment for the Macintosh.

Use REALbasic's visual interface builder and platform-independent language to build native, compiled—not interpreted—professional quality applications in a fraction of the time it would take in C/C++. Because our language is platform-independent you only need to write a single set of code to create applications for both Macintosh and Windows. Leverage your C/C++ experience to extend REALbasic's capabilities using shared libraries, Mac OS Toolbox and Win32 API calls or by writing cross-platform REALbasic plug-ins. Create prototypes, Internet applications, database front-ends, even games with REALbasic! Its OOP language employs everything

you'd expect from a modern development environment—methods, properties, classes, subclassing, inheritance, constructors and destructors, virtual methods, and more. The IDE supports multi-threading, extensibility, TCP/IP controls, plus multimedia and QuickTime tools, and support for standards such as SQL, ODBC, AppleScript, and XCMDs. You can even import Visual Basic forms and modules.

Go to www.realbasic.com NOW to download a FREE trial version or call 512.263.1233.





2000 Runner-Up - Best Macintosh User Experience

*Free update for all owners of REALbasic 2.0 and above. REALbasic and the REALbasic logo are trademarks of REAL Software, Inc. Apple and the Apple logo are trademarks of Apple Computer, Inc., registered in the U.S., used with permission. All other trademarks are the property of their respective owners.

By John C. Welch

Networking the OS X Public Beta

How well does OS X fit into the networked world

WELCOME

In our last look at the OS X Public Beta, we did a general overview of the operating system, and looked a little at networking, mostly at the settings in NetInfo Manager. This time, we'll focus almost entirely on how the Public Beta fits in to other networks, and connects to other Operating Systems.

One of the disadvantages to trying to really exercise the networking features of a beta is the lack of documentation. This has long been one of the real problems with being a Mac OS network administrator, lack of Apple documentation. Almost none of my reliable resources for administering Mac OS networks come from Apple. Instead they come from a host of web sites, Developer TechNotes, third-party books, and 16 years of supporting Macs. For OS X, this needs to change. Yes, third parties will always be a supplement to Apple, but I, and my colleagues in the administration field need to be able to get a complete set of documentation to NetInfo, OS X and NFS, etc. from Apple. In Apple's defense, the deal they have struck with FatBrain.com is a step in the right direction, but as of yet, I have only seen developer documentation come out of it, and Apple has always tried to be conscientious here. There needs to be a similar flood of administration and other similar documentation, and it needs to start happening before the full release, so that the day OS X goes into full release, I will have had access to the documentation I need to begin implementing it that day. If I have to wait a month, 2 months, etc. then in addition to that delay, there will be another month or so delay while I familiarize myself with the documentation, and figure out how I can use it to set up Mac OS X for my network. Most of my fellow administrators will probably be doing the same.

Luckily, for now, there is a decent amount of documentation from such sources as the Omnigroup's mailing lists, and various web sites such as MacNN, etc. But that will not be adequate for long, and with OS X, Apple can not rely on the Mac community to make up for a lack of manufacturer — provided documentation. I don't think they will, but a reminder can't hurt.

NETINFO

You cannot talk about networking OS X without delving into NetInfo. In fact, you cannot talk about administering OS X without dealing with NetInfo. So what is NetInfo? Well, according to Apple's Tech Info Library Article 60038:

"NetInfo is a hierarchical distributed database that is used to keep track of administrative data in NeXTSTEP, OpenStep for Mach, and Mac OS X Server. It can store information on user and group accounts, e-mail configurations, NFS (network filesystem), printers, computers and other resources. Since this information is stored in NetInfo these resources are easily configurable, and can easily be shared over in a network environment."

In other words, NetInfo is how a Mac OS X administrator would maintain their OS X - based network, both at the machine level, and at the network level. NetInfo is also a hierarchal domain model, which, at it's simplest, means that you can have one network that binds together many separate and even independent subnetworks, or subdomains into one organized super-network. With this model you can have, theoretically and unlimited number of subdomains in a root, or '/' domain, ('/' is the NetInfo indicator of the root of all domains on a given NetInfo network.) The machine that is running the / domain is the domain controller. An example of this is shown in **Figure 1**.

John Welch <jwelch@aer.com> is the Mac and PC Administrator for AER Inc., a weather and atmospheric science company in Cambridge, Mass. He has over fifteen years of experience at making computers work. His specialties are figuring out ways to make the Mac do what nobody thinks it can, and showing that the Mac is the superior administrative platform.



Sanctuary.

Move to **digital.forest**. The leading Internet Service Provider for the Macintosh community.

You need maximum flexibility, security and performance from your ISP. You'll find it with digital.forest. We're the proven provider of Macintosh Internet service.

Founded in 1994, our specialty is Mac systems and technologies. In fact, digital.forest is the world's largest Macintosh Hosting Provider. And our hard-earned knowledge of Mac server software is the broadest in the business.

We pioneered Macintosh Server Colocation and FileMaker Pro database hosting services. And our commitment to service keeps growing. So whether you choose us to host your files or house your servers, you've chosen the best.

Go with the industry leader. Discover the sanctuary of digital.forest.

digital.forest



8 7 7 - 7 2 0 - 0 4 8 3

info@forest.net

www.forest.net

Call toll-free in the U.S. and Canada.
For international inquiries, please call 425-483-0483.



FileMaker Pro is a registered trademark of FileMaker, Inc. The digital.forest logo is trademark or registered trademark of digital.forest, Inc.

© digital.forest, Inc. 2000. All rights reserved.

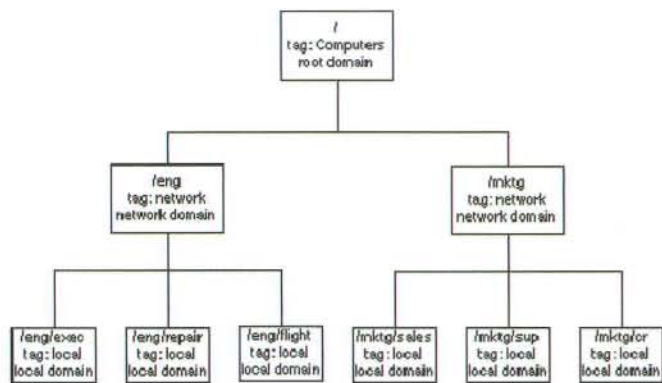


Figure 1.

As you can see, the / domain, named Computers is the root domain. It has access to all other hosts/computers in that domain. The next levels are subdomains: eng and mktg, or in proper notation /eng and /mktg. These are both network domains, under the control of /, but independent of each other. They also have hosts below them. Each of the hosts in the /eng and /mktg domains is an individual computer, or local domain. This local domain can sometimes cause confusion, but there's a reason for it. In many cases, the permissions for a person on their individual computer may need to be different than their permissions on the network. You may want this person to be able to say, change some personal parameters such as screensaver settings on their local machine, but not for the / domain controller. So by having a separate local domain for each machine, you can differentiate between domain and local permissions. (The correct NetInfo terminology in our example would be that / is the parent for /eng and /mktg, which are its children. The hosts in those domains are the children of /eng and /mktg, and the grandchildren of /. I don't really like to use this terminology, as it can get unwieldy quickly, so I go with domain and subdomain. Both work.)

Each domain controller, regardless of level has a NetInfo database that has various information for that domain only, such as users, groups, machines in that domain, services running, printers, etc. They also know who the domain controller above them is. However, each of these domain controllers is separate from any other domain controller at the same level. So, while I may have administrative permissions in /eng, I may only be a normal user in /mktg. The / controller supercedes all subdomain controllers, so if I have administrative privileges for the / domain, then I have them in both /eng and /mktg.

This separation of domains is important. This way, a network admin can limit / admin permissions to only those who absolutely need it, and give /eng admin privileges to another person who needs them for that domain, yet not for /mktg. This allows for easier maintenance of the subdomains, by allowing for smaller sizes, without making them so independent as to be uncontrollable. This also allows for easier user maintenance, as then you only need to enter user information in once. This is due to NetInfo being able to replicate needed information to not only the subdomain controllers, but to other computers that act as backup domain controllers. Domain separation also allows for easier management of printers, users, or any other resource that is a part of the domain.

Even on a standalone machine, NetInfo is what is running much of the basic functions of the OS. Things like services, network settings, user settings, etc. If you look at NetInfo Manager, you'll eventually find the results of almost any preference setting you can imagine.

This is also where a lot of confusion regarding NetInfo starts. NetInfo Manager is not designed to be a general use NetInfo editor. In other words, it's a good place to change NetInfo settings, but not such a good place to create them from scratch. Ideally, NetInfo settings are set and changed from other applications, such as the System Preferences control panels for system settings, Multiple Users for users settings, Print Center for printer settings and so on. In other words, it's a somewhat organized warehouse for preferences. The NetInfo Manager allows you to change and set not only individual behaviors, but overall NetInfo domain settings as well. Unfortunately, the name lends itself to thinking 'Oh, this is where I do all my NetInfo work.', and this is not normally the case. There are occasions where you have to do this, but ideally, you, or someone would write an application that collects input from a user in a user friendly manner, and then uploads that information to NetInfo. This is the safe way to do it, and the recommended way.

I finally figured this out when after six months of working with the folks on the Omni CONTACT_Con-4AFFE0E22 OS X Server Admin list, (an excellent resource by the way, and I highly recommend it to anyone wanting to use the OS X Public Beta in a networked environment. Subscribe at <http://www.omnigroup.com/community/maillinglists/>), and hunting down links on the web, just why there was no real information on using NetInfo Manager. The reason for this is because you're not supposed to. Even older NeXT documentation talks about getting things done in terms of other applications, not NetInfo Manager. This is good, because it is trying to keep people from mucking about in NetInfo, and really, truly destroying their operating system. As an analogy, doing things in NetInfo manager, without knowing what you are doing and why, is about as hazardous as using ResEdit on your System file without knowing what your are doing and why. It's 'a bad thing'.

APPLESHAREIP

This is an obvious one to go over, as it should be the easiest and most completely supported networking model in the Public Beta. Should be, but tends not to be. The most obvious situation is the lack of support for 'pure' AppleTalk. This is not as much of a problem as one would think, as in most cases, networks that are 100% AppleTalk only are rare. The other issue is that the Public Beta doesn't use the Name Binding Protocol, or NBP.

NBP is how the Chooser allowed you to see network devices. When you selected AppleShare, and a zone, if any, in the Chooser, your Mac sent out a NBP request to that zone, or network if zones aren't used. Any AppleShare devices that are able to respond correctly to an NBP request send back a reply that contains, among other things like type, and (sometimes) zone, the server name. The server name is what shows up in the Chooser. Since the Public Beta doesn't support NBP, it does not show up in the Chooser. Nor can it see, in the OS X network browser, any NBP — only servers.

Instead of NBP, the Public Beta uses Service Locator Protocol,

or SLP to discover network resources. Examples of SLP - enabled servers are AppleShareIP (all file-sharing services), Mac OS 9's TCP/IP file sharing, and ExtremeZ - IP from Group Logic. SLP is an internet standard, and Apple was the company that approached the IETF with the idea of SLP as a way to bring over the ease of use to the TCP/IP world that NBP had given the AppleTalk world. This is not to say that a Public Beta machine is unreachable without SLP, it just isn't browseable. You can still connect to it via the AFP URL mechanism, i.e. `afp://thismachinenameorIPaddress`, or the server IP address button in the Chooser.

Other changes to the Apple Filing Protocol, (AFP) between versions 3.0, (the Mac OS X version), and AFP version 2.2, introduced in AppleShareIP 5.0 include:

- Support for longer pathnames and pathnames with Unicode characters
- Support for Unix privileges
- The ability to check for Unix privileges on the server
- Checking to see if the server supports directory services
- Larger file sizes and disk drive sizes
- Authentication method queries
- Log in directory name queries
- Mapping Unicode user and group names to user and group IDs
- Reconnections
- Longer attention messages

So even though pure AppleTalk is not supported, OS X is not cut off from AppleShare networks. But if you still have AppleTalk - only networks, or even LocalTalk networks, now is the time to start moving them to a TCP/IP - based network. OpenDoor, at <http://www.opendoor.com> has not only excellent products to help you do this, but excellent information on ways to accomplish this with as little pain as possible. In addition, there is an article in the January, 2000 MacTech, "Fitting the Desktop into a TCP/IP Environment", that describes some real - world adventures, (namely my own), in doing just that.

NIS

NIS, or Network Information Services, is the network management/directory service primarily used in networks based on Sun Microsystems's Solaris operating system. Like NetInfo, NIS can be used to track user information and privileges, host information, and other similar information. Sun has also developed a version of NIS, called NIS+, which allows for data encryption and larger numbers of users and hosts. Sun is pretty much the only user of NIS+, and OS X cannot be used with NIS+, so we deal with NIS.

NIS is directly supported in the Public Beta, and OS X Server as well, although Apple doesn't directly support NIS itself. (translation: NIS should work, the support is there, but if it doesn't, don't call us.) Also, a simply excellent and clearly - written guide to NIS and OS X is available at <http://www.bresink.de/osx/nis.html>, and goes into step - by step detail on using NIS on the Public Beta, including such things as configuration, troubleshooting, and even automount issues. Configuring NIS is somewhat tricky, and is not for someone who is unfamiliar and uncomfortable with the command line, and NetInfo. As well, the NIS implementation has one rather major error in that it is fairly static. It assumes that you are always going to connect to an NIS network.

This is best shown in the `hostconfig` file, found in `/etc/hostconfig`. (Actually, it's in `/private/etc/hostconfig`. `/etc` is a link, or alias, to `/private/etc`.) Normally, before enabling NIS, the services section looks like this:

```
# Services
AFPSERVER=-NO-
APPLETALK=-NO-
AUTODISKMOUNT=-YES-
AUTOMOUNT=-YES-
CONFIGSERVER=-NO-
IPFORWARDING=-NO-
MAILSERVER=-NO-
MANAGEMENTSERVER=-NO-
NETBOOTSERVER=-NO-
NISDOMAIN=-NO-
MACHIPC=-SECURE-
TIMESYNC=-YES-
QTSSERVER=-YES-
WEBSEVER=-YES-
SERVER=-NO-
```

To enable NIS, we change the `NISDOMAIN` line value from `-NO-` to the name of an NIS domain, i.e. `ournisdomain.nis`. Then, at boot time, the Public Beta machine searches for the NIS server for that domain, and connects to it, so that at login, you can use the NIS services available to you.

The problem arises when you aren't in that NIS domain, or on a PowerBook that isn't on any network. The boot halts at the NIS server lookup, and never times out or otherwise moves on. This obviously needs to be fixed in a way that doesn't require constantly re-editing the `hostconfig` file. As well, OS X does not work well with the standard NIS automounter.

Automounting is the process where, upon login to a machine on an NIS network, certain network drives are automatically connected, via NFS to that machine. This can be things like shared

Engineer for Hire

Henry Carstens
Contract Software Engineer

Cross-platform C, C++
CodeWarrior, PowerPlant
Classic, Carbon, OSX

13 Years Experience
20 Commercial Software Apps Shipped
Fast, Dependable, Self-Motivating

On Site or Off

Henry Carstens
carstens@verticalsolutions.com
503-524-6831

Because how soon you ship *does* matter.

application directories, a user's home directory, etc. There are a couple of ways to do this, AMD and autonfs. Neither of these are supported by the Public Beta automounter. To use NIS automounts, you need to create a BSD-Unix fstab, (file system table). Each line has to be of the format:

```
export mount_point vfs_type options dumpFreq passno
```

The export is the server and path to the exported filesystem. (In unix-ese, exporting a filesystem is the same as sharing a volume in mac-ese.) The mount_point is the local directory on the OS X machine that the export is to connect at. You can enter whatever you like here, it's going to be ignored by the OS X automounter. 'vfs_type' is the file system type. Since this table is only used for network mounts, this is always 'nfs'. The options section is a comma-separated list of mounting options. One of them must always be 'net', which is required for NIS Automounting. Other options are:

- rw: read/write access
- ro: read-only access for all users, including root
- rdonly: same as ro
- suid: set user and set group identifiers will be respected
- nosuid: set user and set group identifiers will be ignored
- exec: binaries from the mounted system can be executed
- noexec: opposite of exec

There are about two dozen or so other options that can be used here. The bibliography at the end of the article has a link that thoroughly explains NIS to NetInfo.

This fstab file must be on the NIS server, and the server's yp-Makefile has to be modified to create the new NIS map. Also, because there has to be an additional key column, the final table will have seven columns.

However, there is still an issue with where those directories will be mounted. By default, the Public Beta puts the directories in /Network/Servers/exporting_servername/path, which is a link from /private/Network/Servers/exporting_servername/path. However, if the network you are on expects home directories to have the path /home/path, then you are going to have some troubles with files programs not working right. The manual way to fix it is to create a symbolic link to the Public Beta directory from a /home directory that conforms to your home directory standards. The other way involves changing the automounter options, found in /System/Library/StartupItems/NFS/NFS. If you are not comfortable with editing Unix configuration files, I would recommend the manual method.

LDAP

Lightweight Directory Access Protocol, or LDAP is emerging as a major player in the network management arena. A non-vendor protocol, it has much of the advantages of NetInfo as far as capability is concerned, but with a much broader base of support. LDAP is domain-based like NetInfo, and can manage users and other resources on a network. It is most widely used as an email addressbook standard. LDAP is not only a product in its own right, but is also supported by the two biggest directory service vendors, Novell and Microsoft.

While the Public Beta supports LDAP, the support is not as full-featured as NIS or NetInfo. This is not a real surprise, LDAP is still a relative newcomer in this arena. The only Apple information on LDAP is in a TIL article, number 24902, last updated in May of 1999. This is actually an article on the lookupd, which is the process that the Public Beta uses to implement things like NIS, NetInfo, LDAP, etc. The TIL article only mentions the basic information that are supported. I have been talking to some people, most notably Luke Howard on OmniGroup's OSX - admin list to try to get a coherent set of instructions on how to implement LDAP support, but it looks to be up to Apple to really get full support for LDAP in OS X implemented properly, as opposed to NetInfo hacks.

The one upside to LDAP support becoming widespread is that, depending on how well this is done, OS X's LDAP support could allow it to play in a network based on Microsoft's Active Directory. Although I have not had the time to test this, since AD supports LDAP, and LDAP connections, in theory, a Mac running OS X could authenticate and participate in an AD domain. If any of you have tried this, please email me and let me know how it worked, or did not work.

WINDOWS NETWORKS

Considering the AD information above, this is limited to non-AD networks. The news here is pretty good, and looks to get better soon. The nice thing is, you can have the Public Beta acting as both a server and a client to Windows PCs. The only bad news is the installation of the related products is not as nice as it could be.

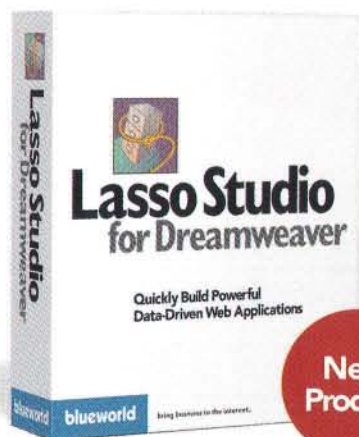
Samba

This is the freeware Server Message Block, (SMB) server that runs on pretty much every version of Unix, including the Public Beta. It allows Unix machines to share resources, files, and printers. There is a client component to Samba, but it is a command line only application, and not the most elegant way to have the Public Beta act as a client. Samba is open-source software, and you can download and build the application, or download it as a pre-built binary. (I downloaded mine from MacNN, <http://osx.macnn.com/features/installsamba.phtml>)

Installing Samba is done from the command line, which is my only real quibble with the application. If someone were to repackage it in a nice OS X installer package, it would be quite nice. In any case, I opened a terminal window, and followed the install instructions on the MacNN page, which are quite simple:

- 1) Download the package to your desktop
- 2) If you have Stuffit Expander 6, then use it to expand the tar.gz file, (the version that ships with the Public Beta will do gz files okay, but gags on tar files.), or from a command line, enter: `tar -xvzf /Library/Desktop/Samba.tar.gz` This will expand Samba, and create the directory on your desktop.
- 3) In the terminal window, enter `su` - (this allows you to act as root, which is needed to install Samba correctly. If you were already logged in as root, this step is unnecessary.) When prompted, enter the root password.
- 4) Enter `cd ~/User/Library/Desktop/Samba` (this line changes directories back to the desktop you were in before you did the

Quickly Build and Deploy Powerful Data-Driven Web Applications



New
Product



New 3.6
Version

Lasso Studio and Lasso Web Data Engine™ Lead The Way.

Building custom and shrink-wrapped database-driven Web applications requires a whole new way of doing things. Having pioneered the Web Data Engine™ over three years ago, Blue World and the Lasso Web Data Engine consistently lead the way providing a feature set Web developers describe as "incredible." Build online stores, discussion forums, resource management systems and other demanding database-driven Web applications with unrivaled performance, ease, security, extensibility, control and flexibility. Develop using multiple languages—including LDML, CDML, Server-Side JavaScript, Java and XML—and deploy across multiple platforms. Your Lasso code works identically regardless to which database you're connected. Lasso solutions for FileMaker® Pro databases easily scale to big iron ODBC-compliant databases like Oracle, Informix, Sybase and more with little or no change. What's more, the Lasso Java Application Programming Interface (LJAPI) provides developers an easy-to-use Java-based API for unprecedented extensibility.

Find out why hundreds of thousands of websites rely on award-winning Lasso technology for their business critical Web data. Download a 30-day evaluation copy at www.blueworld.com/download/ or order securely online today at the Blue World Store at store.blueworld.com.

Lasso Product Line – The leading Web tools for Macintosh and beyond.

bring business to the internet.™

blueworld

- 5) Enter `./SambaInstall.sh` (this says, run this SambaInstall.sh that is in this directory. Because of the way Unix handles paths, you often will have multiple versions of the app, and if your normal path has a version of the same app, without the `./` in front of the application name, you may be running a different version than you think you are.)

Once the install script finishes running, Samba is installed, and after a restart, will be running. However, there is one other part that you will want to install, and that is SWAT, which is a web - based administration tool for Samba. This allows you to configure Samba via a web page, without having to edit configuration files. To install SWAT, you have to reboot so that the Samba processes are running. To check this, open the Public Beta's process viewer application, and make sure that `smbd` and `nmdbd` are running. Open a terminal window, and:

- 1) Enter the `su -` command, unless you are root already.
- 2) Change directories until you are back in the folder you installed Samba from.
- 3) Enter `./SWATInstall.sh` (This runs the SWAT install script)

Once this has finished, you can open up a web browser, either locally, or on any computer that can see the Public Beta mac, and connect to SWAT by using the URL `http://machinename:901` . You will need to log into SWAT as root to do any real configuration setups. As there are hundreds of options available in Samba, depending on your needs, and your network, I will not attempt to even begin listing all of them here, but rather *highly* encourage you to read the online help available in SWAT. It is quite thorough, and gave me all the information I needed to set up my Public Beta machine. The online help will also allow you to configure Samba to plug into existing Windows networks without disrupting that network.

Samba does an excellent job, once set up, of allowing your Public Beta Mac to share files with Windows PCs and other SMB clients such as DAVE, from Thursby Systems. It can act as a domain server for Windows95/98 pcs, but not to NT or 2000 systems yet. That feature is in work, but there is no date on this. This feature, to be part of Samba 3.0, would allow Samba to act as a Windows NT 4.0 Domain Controller, not an Active Directory controller. The Samba team is hoping to integrate Samba into Active Directory domains via Kerberos or LDAP. As always with Open Source products, the more people they have working on it who know what they are doing, the faster it gets done. If you have the skills and the time, by all means, go to <http://www.samba.org>, and sign up.

Okay, we have set up Samba, and Windows folks can see our stuff, but now we want to see theirs. Well, as I mentioned before, there is a command-line utility called `smbmount` that will allow you to do this, but this *is* the Mac OS after all, and while odd things for server applications may be acceptable, on the client side, we want our GUI! Well, fear not, for we have our GUI.

The answer is Sharity, a GUI client for CIFS networking. (CIFS stands for Common Internet File System, and is essentially an improved form of SMB.) Sharity is produced by Objective Development, at <http://www.obdev.at/Products/SharityForOSX.html>. Sharity allows you to browse and mount Windows and Samba server shares within the Public Beta. The install and setup is all GUI based, although a bit odd. You download an installer, which then installs the Sharity installer. To run this installer, you have to be logged into the OS X machine as root. The installer won't work otherwise.

Sharity, like Samba has a lot of options, and I'm not going to go into them all. It is also still in beta, so it can be quirky at times, but when it is working, it is fast, and easy to use. I would like to see Sharity emulate the AppleShare networking in OS X, and mount the shared drive on the desktop as well as in the /Network directory path, as that is what Mac users expect to see on their Macs, and since Apple is continuing this for network shares, other network connectivity vendors should follow suit. Actually, I like the idea of the drive being in both places. For example, with an AppleShare mounted drive, it shows up in both Connected Servers and the Desktop, giving you two methods to access it, depending on your needs.

Another similarity to Samba is that Sharity does have solid online help. Although most of it currently references the Linux versions, the information applies across platforms well enough to get you to where you need to be. In any case, between Samba and Sharity, you can fit the Public Beta into a Windows - based network reasonably well, and have better hardware and software to boot!

Database application generation from AppMaker; d-Base compatible engine; Faircom c-tree Plus c++ interface with many added features; cross-platform graphing; cross-platform report-writer; XML, RTF, HTML output; PowerPlant interface; MFC interface; easy-to-use API; cross-platform file handling and directory iteration; source code; scales to client-server; single and multi-user royalty free; supports millions of records; server and standalone versions run on Mac, Windows & Unix; Database application generation from AppMaker; d-Base compatible engine; Faircom c-tree Plus c++ interface with many added features; cross-platform graphing; cross-platform report-writer; XML, RTF, HTML output; PowerPlant interface; MFC interface; easy-to-use API; cross-platform file handling and directory iteration; source code; scales to client-server; single and multi-user royalty free; supports millions of records; server and standalone versions run on Mac, Windows & Unix; Database application generation from AppMaker; d-Base compatible engine; Faircom c-tree Plus c++ interface with many added features; cross-platform graphing; cross-platform report-writer; XML, RTF, HTML output; PowerPlant interface; MFC interface; easy-to-use API; cross-platform file handling and directory iteration; source code; scales to client-server; single and multi-user royalty free; supports millions of records; server and standalone versions run on Mac, Windows & Unix; Database application generation from AppMaker; d-Base compatible engine; Faircom c-tree Plus c++ interface with many added features; cross-platform graphing; cross-platform report-writer; XML, RTF, HTML output; PowerPlant interface; MFC interface; easy-to-use API; cross-platform file handling and directory iteration; source code; scales to client-server; single and multi-user royalty free; supports millions of records; server and standalone versions run on Mac, Windows & Unix;

OOFILE
is now
Free
www.oofile.com.au

NOVELL

Although there is currently no OS X - native offerings from Novell, that will soon, (according to Novell) be remedied. Native File Services for Macintosh are due to be released in the first quarter of 2001. This will allow for Mac clients to authenticate and access NetWare 5.X and 6 networks and resources. There is no client component for this, it works through Apple's native authentication, and Novell's NMA authentication services. This does allow OS X Macs to use Novell Directories, and the resources managed by Novell Directory Services, (NDS).

What this does not mean is a single signon for Novell Networks. According to information from Novell, Native File Services is a 1.0 release, and will not have single signon capabilities. They are looking to include that in a future release, so anyone currently using Novell networks should talk to their representatives, and let them know how important that feature is to you.

CONCLUSION

Again, we all have to remember that this is a beta, and a beta of a client at that. There are a lot of things we may want from it that are more suitable to a server version of OS X. At the moment, the Public Beta seems to be doing a decent job of fitting in, but that needs to be improved. There needs to be much better support for other networks that aren't NetInfo based. The setups for LDAP and NIS in particular should be streamlined, and put under a proper interface. Yes, some enterprising developer may do this on their own, but OS X is an Apple product, and Apple needs to take the lead here. Apple also needs to work with developers such as Novell and Microsoft, to help ensure a

seamless networking experience for OS X, both as a client and as a server. Having a simple menu to select between different networking systems, and having it work seamlessly once that choice was made would do much to eliminate the last of the old "Macs are incompatible with X's network system" arguments.

Apple should also look into making Samba an integral part of the OS, and even talk to Objective Development, or Thursby about having an integrated SMB client in the OS. By having those capabilities as part of the basic OS, the door is opened to improve upon those basic abilities, but the OS would be able to communicate better, and more easily with the outside world. Again, this is the kind of thing that would really give Apple a universal network client. Considering the level of network in most computer owners homes is getting more complex by the month, Apple should be able to use this the day OS X ships.

The bibliography is a list of URLs that I have found useful in putting this article together. Along with that, I'd like to give a special thank you to everyone on the OmniGroup Mac OSX-admin list. A lot of this article would have been much harder without them.

BIBLIOGRAPHY AND REFERENCES

- Version Tracker's OS X section:
http://www.versiontracker.com/vt_mac_osx.shtml
- The OmniGroup's OS X mailing lists:
<http://www.omnigroup.com/community/maillinglists/>
- Apple's OS X mailing lists: <http://lists.apple.com>
- MacNN's OS X section: <http://osx.macnn.com/>
- Information on NIS and NetInfo:
<http://www.bresink.de/osx/nis.html>

107

Webmasters! Start your web hosting business with edition.net reseller hosting!

dedicated platforms
single and multi-domain hosting
your choice of Mac OS or Linux platforms
FileMaker Pro hosting featuring Lasso
QuickTime streaming server

Our web-based control panels make it easy to
deploy your multi-domain server

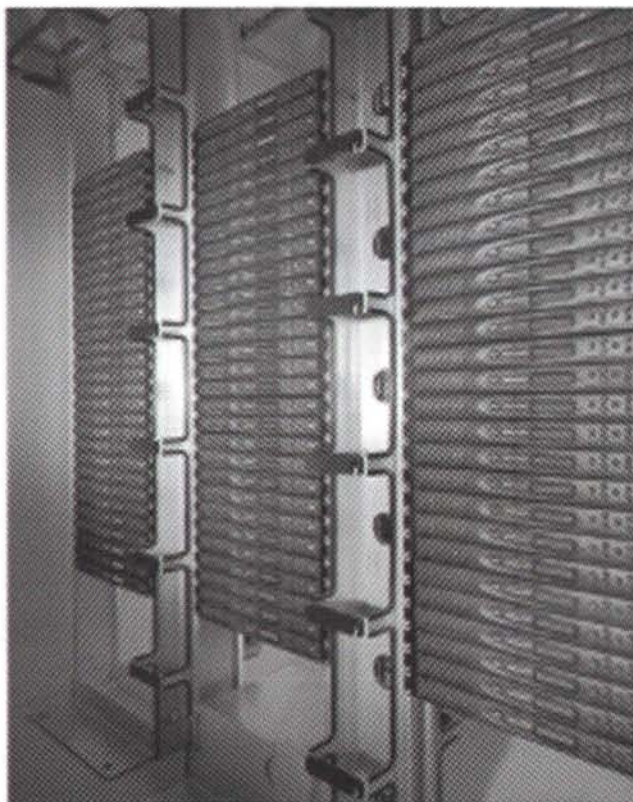
edition.net – the Fine Arts Network

<http://www.edition.net>

info@edition.net

+1 (877) 225-3821 toll free

+1 (310) 379-8563 international





Tools for the Trade

The focus of this article is Apple's development tools. Apple has been working on a Mac OS X tool suite to address all facets of Mac OS X development needs and provide valuable options for all developers.

Recalling from the first article in this series, Mac OS X is all about choices: what to develop, what to take advantage of in this new state of the art operating system. From its UNIX — based kernel to its advanced object — oriented application frameworks, Mac OS X provides a wide variety of opportunities for new products, each of which presents a need for development tools support.

The first developer tool to highlight is Mac OS X reference documentation. In particular, developers wishing to understand and take full advantage of Mac OS X's capabilities should start by

The Apple Developer Connection

(ADC) ensures the success of Macintosh developers by supporting them at Apple events. During Macworld San Francisco, ADC members displayed software and hardware products to customers and received discounts on booth space and CPUs. ADC has similar programs in place for Macworld Tokyo, Macworld Paris, and many other shows.

ADC Members in Europe were recently treated to a series of in-depth developer briefings. These briefings, designed for developers who want to make the move to Mac OS X by carbonizing their applications and plug-ins, covered topics such as Aqua, Carbon, Cocoa, Core OS, and Java.

Be sure to mark your calendar for Apple's Worldwide Developers Conference 2001, which takes place in San Jose, California from May 21-25. Details and preliminary schedules will be available soon.

reading "Inside Mac OS X: System Overview", available from the Apple Developer Connection web site in the Mac OS X Documentation section, System Overview category:

<http://developer.apple.com/techpubs/macosx/macosx.html>

This book will take you through the entire operating system from the kernel OpenSource layer (Darwin) to the Aqua User Experience and give you the system overview necessary to make development choices.

Let's say you've already chosen your development area; you have your product in mind; you know what you want it to do; you have chosen your development API set from the choices available in Carbon, Cocoa, Java, BSD or Kernel centric realms. In front of you is the next set of choices: What development tools will you use to achieve your development goals most effectively?

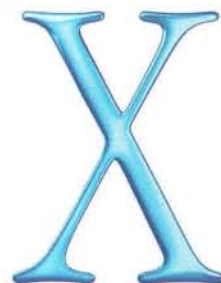
Both Apple Computer and third-party tool vendors have a long history of delivering Mac OS development tools which span the gamut: from high — level, easy to use, and fast to learn tools to commercially oriented, high — power tool suites that approach every kind of operating system component and application development. No less is true for Mac OS X. Apple has been working closely with third-party IDE and other tool vendors to bring their tools to Mac OS X, including Metrowerks' CodeWarrior Pro and Real Software's REALbasic along with many others. For a more detailed listing of available tools and URL links to their vendors, see the Development Tools page on the Apple Developer Connection web site:

<http://developer.apple.com/tools/>

Let's now take a brief look at Apple's development tools for Mac OS X.

Project Builder

Project Builder is Apple's integrated development environment (IDE) for Mac OS X. It is designed to fully support all of the major platform initiatives of Mac OS X, such as the Carbon and Cocoa frameworks, Java, I/O Kit and the application bundle mechanisms. It provides templates for quick creation of new projects; a few clicks of



Godfrey DiGiorgi is Technology Manager for Development Tools and Cocoa in Apple Worldwide Developer Relations. He's in his 8th year at Apple, having spent six years in Developer Technical Support as an engineer and manager. Godfrey has been involved with software development on Mac OS since 1984, working on Radar Imaging applications while at NASA's Jet Propulsion Laboratory and later in the chemical research industry as lead architect/engineer for a cross-platform application framework implemented on Mac OS. He also spent two years at Sun Microsystems as part of the team that developed the specification for the Java Card language platform.

the mouse will create a starter project for an I/O Kit driver, a standard UNIX type tool, or a Cocoa or Carbon application. Everything that can be written for Mac OS X can be created with Project Builder including applications, tools, frameworks, libraries, plug-in bundles, down to kernel extensions and device drivers.

Project Builder provides project editing, search and navigation facilities, file editing, build system and target setup, and high — level debugging facilities. Language implementations allow development using C, C++, Objective-C and Java. CVS source code control system access is integrated into the user interface.

All of these features are built on top of industry standard tool components provided by Mac OS X, allowing for great robustness and compatibility with source code developed on many platforms. C, C++ and Objective-C compilers are the gcc compiler suite; Java is compiled with either **javac** from Sun's JDK or with the **jikes** compiler from IBM. The build system is based upon **jam**, a high performance 'make' with reduced maintenance requirements. Project Builder's debugger uses **gdb** providing both high level debug panes and direct access to **gdb** commands for any desired runtime inspection.

The design centers of Project Builder provide robustness and flexibility along with multi-user, multi-project capabilities. In order to keep project management and debugging clear and consistent, Project Builder uses a paned-window approach in its UI with growable/collapsible areas that allow maximized effectiveness when working in a specific area of its functionality. It can open separate editing windows when multiple files need to be available at the same time. Project Builder is also able to accommodate pre-existing and legacy projects, with support for makefile driven builds as well as CodeWarrior project imports.

Interface Builder

Interface Builder is Apple's user interface design tool for applications. Developers using its graphical editing environment can manage virtually every aspect of creating a well designed user interface that adheres to the Aqua user interface guidelines. It's great for prototyping and fast development changes.

Interface Builder understands all Carbon and Cocoa UI elements and produces 'nib' files that provide a modern way to manage and develop the static resources used by applications. Recent enhancements to Interface Builder include layout tools that use guides and distance measurement tailored for Aqua, a high degree of integration with Project Builder, and improvements in performance, stability, and "Undo" capability.

Interface Builder provides the easiest way to use advanced Carbon Events in the Carbon application development process and it is a highly tuned part of the Cocoa application development process through implementation of the Cocoa concepts of outlets and actions. By leveraging the AppKit and Foundation frameworks

of Cocoa, Interface Builder provides a highly efficient rapid application development environment.

Interface Builder works with Project Builder to make application design and implementation more productive and to create highly reliable, good looking applications. Expect much more to come as Apple evolves these applications to take advantage of upcoming Mac OS X features.

Performance Tools

Mac OS X is a new operating system with new libraries and semantics that can differ in many ways from previous incarnations of Mac OS. There are different costs to the use of these APIs, often different algorithms are required to get the most out of your implementation. Apple has developed several performance analysis tools that are available to you now. Inspection capabilities that we have developed fall into roughly three categories:

- System State: memory, resource usage, system calls
- Execution Patterns: profiling and tracing
- Memory usage: tracing allocations and leaks

One group of these tools leverages the command line environment provided by BSD services and allows for minimally invasive ways to gather data about a running application. These include but are not limited to:

top - determine what tasks are executing and statistics
fs_usage, **sc_usage** - observe system call usage

At a higher level, there is **Sampler** for 'on the fly' profiling. This tool stops the program at intervals to see what's running,

Discounts on Mac OS X Migration

Bring Your Apps to Mac OS X

With Mac OS X Public Beta here, and the commercial release just over the horizon, Mac developers everywhere have a huge need for Carbon and Cocoa application porting, Aqua user interface implementation, and driver development services. Several high-quality software engineering firms, in association with the Apple Developer Connection, are offering these services at very attractive discounts to all ADC Select and Premier members.

<http://developer.apple.com/mkt/macosexmigration.html>

summarizes which functions and call traces were seen, and provides a statistical look at runtime operation. It requires no recompilation or source code changes to do its job. It's best for exploring an application's execution in large context; the command-line version, 'sample', is great for seeing why an application is hanging.

A last tool to highlight here is MallocDebug. MallocDebug is intended to do memory heap analysis, answering difficult questions like:

- How much heap memory is a program using?
- Where has a buffer been overrun or underrun?
- Which functions request large chunks of memory?
- Am I leaking memory?

MallocDebug can tell you how much memory is used at specific times and identify the call stack leading to each allocation. It can also see all Carbon, Core Foundation and Objective-C memory allocations.

Use of these analysis tools can contribute to substantial improvements in application performance and reliability.

Delivery and Documentation ... where to go for more!

All of Apple's Mac OS X development tools are distributed on a companion CD to Mac OS X called the "Mac OS X Developer Tools CD". This jam-packed CD contains packages that install the applications, reference documentation and sample code into a Mac OS X configured system.

Documentation for tools and APIs is available from within running applications using the Mac OS X help system's "Developer Center" and in the form of HTML pages viewable with any browser. On a Mac OS X Public Beta system with the Developer Tools CD installed, see the file:

/Developer/Documentation/DeveloperTools/devtools.html

The first version of the Mac OS X Developer Tools CD was delivered in conjunction with Mac OS X Public Beta, mailed directly to all Premier, Select, and Student members of the Apple Developer Connection. A downloadable version was made available to all ADC Online program members. Future plans include similar distribution, and web-based component and documentation updates. Stay tuned for future announcements on these plans!

Summing up

Choosing your development tools is important. You should weigh the strengths of all the development tool options to achieve your implementation goals. Apple's tool suite provides a rich set of capabilities for Mac OS X development allowing you to access every feature of this new operating system and providing you with the broadest development flexibility.

Did You Know?

Finding API Information from Project Builder

Through integration with the Mac OS X help system, Project Builder makes it easy to look up reference documentation on functions, methods, structures, and other API symbols directly from your code. This feature is easy to set up and use.

1. Index your project. Choose Index Project from the Project menu. Indexing caches the definitions of all project symbols, including those of imported frameworks and libraries. These symbols are also captured in the reference documentation as special tags embedded in the HTML.

2. Search for definitions. Choose Show Batch Find from the Find menu (or click the Find tab) and, in the Batch Find pane, choose Definitions from the pop-up menu. Then type the name (full or partial) of an API symbol in the Find field and click the Find button. To look up a symbol directly from your code, select the symbol in your code and press Command-Shift-E (Find > Find Selection).

3. Click the book icon. When you search for a symbol definition, Project Builder returns a list of all symbols that match what you entered. If documentation on the symbol is available, it displays a book icon next to the symbol. Click the icon to display the documentation in Help Viewer.

The Developer Help Center section of Help Viewer includes a feature allowing you to copy declarations and other code "snippets" into your project. Click the folder icon next to the declaration or example to copy it to the Clipboard. Then, in Project Builder, move the insertion point to the desired place in your code and paste the declaration or code example from the Clipboard.



New Mac OS X Related Releases

The following software is available from the Download Software area of the ADC Member Site at:

<http://connect.apple.com/>

- **Latest CarbonLib SDK**

The CarbonLib SDK provides all the files needed to begin Carbon development.

- **Apple MacApp 15d3**

Apple MacApp 15d3 is an application framework for creating user-friendly, object-oriented applications for the Mac OS. This release includes support for building with Project Builder, a new framework of helper classes for Mac OS X's Core Foundation, and updates to a variety of Toolbox suites in ACS.

Developer Documentation

The following new and updated documentation is available to help you on your way to successful Mac OS X application and peripheral development at:

<http://developer.apple.com/techpubs/>

- Moon Travel Tutorial (Building a Simple Carbon Application)
- Providing User Assistance With Apple Help
- Handling Unicode Text Editing With Multilingual Text Engine
- Aqua Human Interface Guidelines
- Navigation Services for Carbon API Reference
- Managing Fonts With the Font Manager
- Programming With Language Analysis Manager
- Using Apple Japanese Analysis Engine and Access Method
- Programming With Dictionary Manager
- Rendering Library Preliminary Reference
- Carbon Specification
- Carbon Event Manager Preliminary API Reference
- Carbon Porting Guide
- Drag Manager Reference
- AGL Reference
- Games Sprockets Reference
- Power Manager Reference
- Matching Fonts With Font Sync
- Window Manager Reference
- Dialog Manager Reference
- Menu Manager Reference
- File Manager Reference
- Transferring Data With the URL Access Manager
- Implementing Security Features Using the Keychain Manager

- TN2009 - The Browser Control (aka That ListView Thing)
- TN2006 - MP-Safe Routines

- TN1128 - Understanding Open Transport Memory Management
- Q&A JAVA28 - Creating JNI Libraries With Project BuilderMac
- Q&A JAVA29 - Creating Mac OS X Applications From Java JAR Files
- SAMPLECODE - Networking: OTMP
- SAMPLECODE - Networking: DumpNetworkSetup
- SAMPLECODE - Overview: MoreIsBetter
- SAMPLECODE - Processes: CallMachOFramework

Carbon Tips & Tricks

The Apple Developer Connection web site now includes a Carbon "Tips & Tricks" web page dedicated to sharing special techniques for debugging and building Carbon applications.

<http://developer.apple.com/macosx/carbon/tipsandtricks.html>

Upcoming Seminars and Events

For more information on Apple developer events please visit the developer Events page at:

<http://developer.apple.com/events/>

Training and Seminars

Programming with Cocoa

Taught by Aaron Hillegass at the Big Nerd Ranch, Ashville, NC and Atlanta, GA. Five-day classes are taught on developing web-based and Mac OS X applications

<http://www.bignerdranch.com/when.html>

Developer Related Conferences

- **MACWORLD Expo Tokyo 2001**

February 22-24

Macworld conference and Expo Tokyo provides you with the ideal venue to introduce your company and products to over 200,000 eager attendees. Special exhibitor packages and hardware discounts for ADC Premier and Select members are available.

<http://developer.apple.com/mkt/mwtokyo2001.html>

- **Worldwide Developers Conference (WWDC) 2001, San Jose, California**

May 21-25

WWDC 2001, the definitive event for Mac software and hardware developers, is coming to the San Jose Convention Center a week later than usual this May. Mark your calendar now and read ADC Direct for updates.

<http://developer.apple.com/mkt/>

The products you need, with the prices and service you deserve... guaranteed.

Power Tools for Programmers!

CodeWarrior Pro 6

NEW!

CodeWarrior for Mac OS is a powerful integrated development environment that includes a fully object-oriented application development framework called PowerPlant. With CodeWarrior for Mac OS you can quickly develop reliable, professional quality applications that execute on Classic Mac OS, or OS X.

NEW Version

ONLY! \$359

Spotlight

Spotlight is the first Macintosh "Automatic Debugger". It can automatically locate run time errors in your code and display the offending source code line. Unlike similar tools on other platforms Spotlight is easy to use. No source code changes are necessary for application debugging. Spotlight can automatically check for wild pointers, memory leaks, overwrites, underwrites, invalid dereferencing of handles, and even toolbox parameter validity checking -- spotlight knows Macintosh verifying parameters to over 400 toolbox calls.



\$189

Resorcerer 2.2

Resorcerer is the only supported general-purpose resource editor for Macintosh. Relied upon by thousands of Mac developers, Resorcerer features a wealth of powerful yet easy-to-use tools for easier, faster, and safer editing of Macintosh data files and resources. Whether you have to parse a picture, debug a data fork, design and try out Balloon Help, create a scripting dictionary, create anti-aliased icons, design and edit a custom resource with 40,000 fields in it, create C source code to run a dialog, or any of hundreds of other resource-related tasks, Resorcerer's magic will quickly save you time and money.



\$239

REALbasic 2.1

NEW!

REALbasic is the programming tool for "the rest of us." Each window type, control, and menu is preconfigured and instantly works as it should. The drag and drop Window Editor allows you to quickly and easily create your application's interface so you can focus on the important part--your creativity. Includes 900 page manual, examples and tutorial on CD ROM! The professional version has all the power of the standard, plus database support and allows you to cross compile your code for Windows with a single click!

As low as

\$139

VOODOO Server

VOODOO Server is a version control system for software developers using Metrowerks CodeWarrior under Mac OS. VODOO Server and the corresponding VODOO clients (the included CodeWarrior VCS plug-in and the VODOO Admin application) are designed to offer reliable and robust version control features while minimizing the administrative overhead that usually accompanies version control. If you're a single programmer or managing a team of developers, version control can make or break your project. Do it right, with VODOO.



\$79

Future BASIC 3

NEW!

One of the most flexible and powerful development environments on the Macintosh today! Easily create programs with the visual program editor, drop into the BASIC editor to define powerful logic with the worlds easiest programming language, or work directly with the Macintosh toolbox. The only BASIC compiler on the market that gives you 100% access to all of the power of the Macintosh toolbox!



\$159

and hundreds more!

<p>PageCharmers 2.0 \$139</p>	<p>Scripter 2.0 \$179</p>	<p>Script Debugger \$189</p>	<p>ToolsPlus Lite \$99</p>	<p>FaceSpan 3.0 \$179</p>	<p>WebSpice 1,000,000 \$89</p>	<p>WebSpice Animations \$89</p>	<p>MkLinux \$39</p>	<p>PowerKey Rebound \$89</p>
---	---	--	--	---	--	---	---------------------------------------	--

Developer DEPOT®

PO Box 5200 • Westlake Village, CA • 91359-5200 • Voice: 800/MACDEV-1 (800/622-3381)
Outside US/Canada: 805/494-9797 • Fax: 805/494-9798 • E-mail: orders@devdepot.com

www.devdepot.com

Master the Web!

WebSTAR Server Suite 4.3

WebSTAR Server Suite is a complete set of powerful and easy-to-use Internet servers for the Mac OS. Effortlessly serve web pages, host email accounts, publish databases, and share files - all with a single application on one Mac! WebSTAR Server Suite is perfect for Internet or Intranet serving, single or multiple sites, small and large businesses - it's power and ease-of-use saves any organization time and money.



\$539

CyberGauge 3.0

Monitor the bandwidth usage of up to five different machines on your network! Do you need to upgrade your webserver? How hard is your eMail server working? Are you getting all the bandwidth you're paying for? Not only can CyberGauge answer all these questions, new features allow CyberGauge to eMail or page your network device becomes unresponsive or passes a threshold of usage you define - an essential first line of defense for early detection of denial of service attacks and necessary for warning you and tracking quality of ISPs that may have brown outs and shutdowns.



\$249

Funnel Web Pro

Funnel Web is the ultimate web analysis solution for professionals. Specifically designed for profiling web site usage and monitoring customer usage patterns, Funnel Web is ideal for examining server performance and online effectiveness. Funnel Web can analyze log file formats from any server, WebSTAR, WebTen, even Unix or NT hosted servers. Discover the most popular pages on your site, track server loads & optimize server performance, profile visitors based on organization, domain name, country, browser, etc. Funnel Web does it all!



NEW!

As low as **\$329**

NetBarrier

NetBarrier offers a Personal Firewall, Antivandal protection, and Internet Filtering. Protect your machine from intrusions by Internet or AppleTalk. Incorrect passwords and individual actions are logged, you are alerted to hostile actions, and intruders are easily isolated. Internet Filtering allows you to be sure that passwords, credit card numbers, and other sensitive information can never be exported from your computer - the content itself is filtered before any transfer! (Rated 4 mice by Macworld Magazine)



\$49⁹⁵

...and great hardware solutions!

2 USB PCI Card

Add two USB ports to your older Macintosh. Connect up to 127 devices to the Universal Serial Bus (USB) that is Apple's new standard for desktop connectivity. USB mouse devices, keyboards, joysticks, game controllers, printers, scanners - connect them all to your current computer. Installs in minutes!



\$32⁹⁵

Macsense USB Full Size Keyboard

Just plug this keyboard into your Mac and start typing! The UKB-600 keyboard from Macsense is designed to get you typing quickly and easily, without any hassle or compatibility worries. It features two tone translucent design, colored to match your favorite flavor of Macintosh. It offers soft touch with positive tactile feedback and build built in USB port on either side of the keyboard. Includes a 5' USB cable and is 100% Macintosh compatible, simply plug and play, as easy as Macintosh!



\$44⁹⁵

Dr. Bott Moni Switch ADB or USB

Do you need 4 monitors and 4 keyboards for your 4 servers? With Dr. Bott Moni-Switch you can connect a single keyboard and monitor to up to 4 machines at once! A simple flick of a switch directs the video input and keyboard commands to the appropriate CPU! Available in USB and ADB models, with 2 or 4 machine support, and bundles with USB PCI cards so you can mix and match USB and ADB machines with the same Moni-Switch! Great for programmers to do back ground compiles, ideal for server rooms overcrowded with monitors and keyboards!



As low as **\$129⁹⁵**

Macsense Internet Sharing Router

Looking to get your whole office online without shelling out thousands of dollars? If so, the XRouter Internet Sharing Hub offers the perfect solution. This amazing Ethernet-to-Ethernet hub connects an entire network of up to 252 users to the Internet using only one ISP account and one Cable or DSL modem!



\$199

By Tim Monroe

Honey, I Shrunk the Kids

Using QuickTime's Standard Image Compression Dialog Component

INTRODUCTION

In a previous *QuickTime Toolkit* article ("Making Movies" in *MacTech*, June 2000), when we built our very first QuickTime movie, we used a couple of Image Compression Manager functions to compress each video frame so that it (and hence the entire movie) took up less space on disk. The size reduction was significant: simply adding 100 uncompressed frames to the movie would have resulted in a movie file that was about 12 megabytes in size. Using JPEG compression, we were able to reduce the final movie file size to about 470 kilobytes.

In that article, however, we cut some corners by hard-coding the compression type when we called `GetMaxCompressionSize` and `CompressImage`. It would have been nice to provide the user with a choice of compression algorithms, and indeed perhaps even an indication of what any particular compression algorithm would do to the penguin images. Happily, QuickTime makes this very easy to do, by supplying the *standard image compression dialog component*. We can use this component to perform two main tasks. First, as the name suggests, we can have it display a dialog box in which the user can adjust compression settings for a single image. **Figure 1** shows the *standard image compression dialog box*.

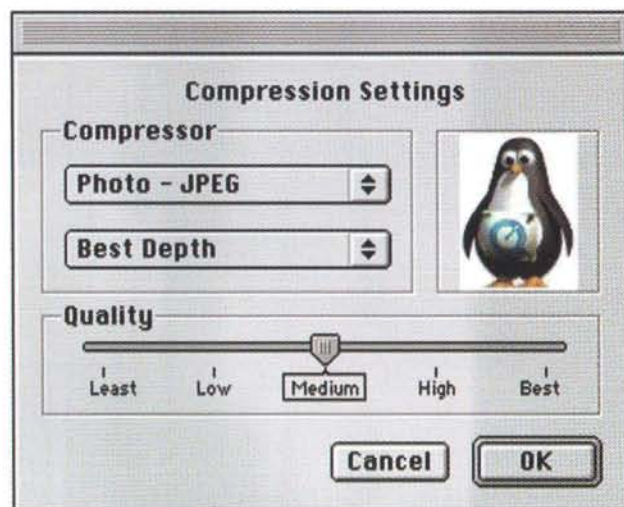


Figure 1: The standard image compression dialog box.

The standard image compression dialog box contains a pop-up menu that lists the available image compressors. It also contains a pop-up menu that lists the available pixel depths supported by the selected compressor. Finally, the dialog box contains a slider control for adjusting the image quality. As the user varies the compressor, pixel depth, or image quality, the standard image compression dialog component adjusts the thumbnail picture to show what the image would look like if compressed using the selected settings.

The second main task that the standard image compression dialog component can perform is to compress the image. That is to say, it can not only retrieve the desired compression settings from the user, but it can also do the actual compression for us (thereby saving us from having to call `GetMaxCompressionSize` and `CompressImage`). For this reason, the component is sometimes also called the *standard compression component*. **Figure 2** shows the result of using the standard image compression dialog component to compress our penguin picture using the PNG image compressor at 16 levels of grayscale and the highest available quality.

Tim Monroe works in the QuickTime Engineering team at Apple. You can contact him at monroe@apple.com.

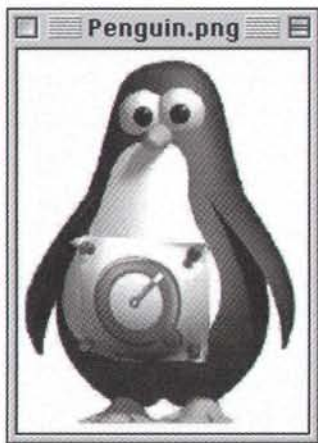


Figure 2: The penguin picture compressed with PNG grayscale.

In this article, we'll see how to use the standard image compression dialog component to elicit compression settings from the user and to compress images using the settings selected by the user. We'll also see how to use the standard image compression dialog component to compress a *sequence* of images (for example, the sequence of images that make up our penguin movie). We'll begin by taking a more focused look at compression itself. While the basic idea is straightforward, there are a handful of concepts we'll need to understand before we can start using the standard image compression dialog component. Then we'll spend the rest of this article investigating the compression-related parts of this month's sample application, QTCompress. The Test menu for QTCompress is shown in **Figure 3**; as you can see, it contains only one menu item, which compresses the image or sequence of images in the frontmost window.



Figure 3: The Test menu of QTCompress.

COMPRESSION

Compression is the process of reducing the size of some discrete collection of data, presumably without unduly compromising the integrity of that data. The basic goal, of course, is to be able to store the data in less space and to use less bandwidth when transferring the data over a network. Particularly for multimedia content like large color images, movies, and sounds, uncompressed data (also known as *raw data*) simply takes up too much space on disk or too much time to transfer over a network. It's almost always better to store and transfer compressed data, which is then decompressed during playback.

In QuickTime, compression and decompression are handled by components called *codecs* (which is short for compressor/decompressor). The available codecs effectively

Always Thinking

Professional Macintosh & Internet Development

Always Thinking's professional developers will help you meet your Macintosh and Internet deadlines! So if you're...

- on a tight deadline and need additional talent
- losing valuable development time debugging
- having trouble finding good developers

... **Always Thinking's** team of experienced programmers will provide you with a timely and affordable solution.

We deliver more than code — a complete project. Our software engineers work with you to:

- Create clear, solid project specifications
- Design and develop your application or web site
- Tune and optimize your software's performance
- Thoroughly test your application or site
- Completely document your project
- Provide training to your team

Commercial Product Development

Do you have an exciting idea for an application? Turn to **Always Thinking** to make it a reality. We have firsthand experience developing and shipping award-winning commercial applications for our clients and our own Thinking Home, a 2000 Apple Design Award winner.

Web Site Design & Development

Get a sound e-commerce system tailored for both your immediate needs and long-term growth. Our engineers can develop the Internet applications to transform your company into an e-business.

Successful web sites are more than graphics and code. We have the Internet marketing know-how to ensure your site is an effective business tool.

Realize substantial savings by moving to online pre-sales information, ordering and support.

Tell us about your project, toll-free

(800) 556-9559

(843) 986-0308



Always Thinking, Inc.
27 James Byrnes Street
Beaufort, SC 29902

www.alwaysthinking.com sales@alwaysthinking.com

define the kinds of compressed data that QuickTime can handle. Apple has written a large number of codecs itself and also licensed some other codecs from third-party developers. Ideally, it would be nice if QuickTime supplied both a compressor and decompressor for every kind of data that it can handle, but sadly that isn't the case. For instance, QuickTime can decompress and play MP3 files, but it does not include a component that can compress sound data into the MP3 format.

For the present, we'll be concerned primarily with compression and decompression of images and sequences of images. In this case, there are two basic kinds of compression: spatial compression and temporal compression. *Spatial compression* is a means of compressing a single image by reducing redundant data in the image. For instance, our penguin picture has large areas of pure white; a good spatial compressor would encode the image so as to avoid having to store a 32-bit RGB value for every one of those white pixels. Exactly how the encoding is accomplished varies from compressor to compressor.

Temporal compression is a means of compressing a sequence of images by comparing two adjacent frames and storing only the differences between the two frames. It turns out that many common sorts of video change very little from frame to frame, so a significant size reduction can be achieved by storing a full frame of the video and then the subsequent differences to be applied to that frame in order to reconstruct the original image sequence. In QuickTime, the full frame of video is called a *key frame*, and the subsequent frames that contain only the differences from previous frames are called *difference frames* or *delta frames*. (Other media technologies use other nomenclature. Key frames are also called *intraframes*, and difference frames are also called *interframes*. MPEG calls key frames *I-frames* and has two sorts of difference frames, *B-frames* and *P-frames*.)

Figure 4 shows a key frame (on the left) and the immediately following difference frame (on the right). The difference frame is pretty worthless when viewed by itself; but when the decompressor, using its special algorithms, applies the difference frame to the key frame, we get the actual image displayed in the movie, shown in **Figure 5**.

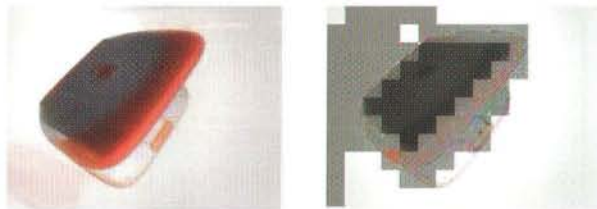


Figure 4: A key frame and the following difference frame.

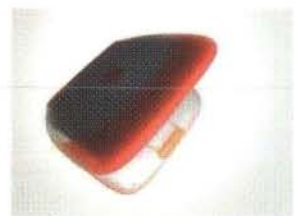


Figure 5: The movie frame as reconstructed by the decompressor.

In this particular case, the space savings are significant. In the movie file, the key frame occupies 4076 bytes, while the difference frame occupies only 3120 bytes. The next difference frame occupies only 1184 bytes. The kinds of space savings you'll see depend of course on the actual content of the movie. For instance, a music video with lots of fast cuts and motion will not compress as well temporally as a movie of paint slowly drying.

In theory, a temporally compressed movie could consist of a single key frame followed by a large number of difference frames. But in practice, key frames are interspersed throughout the movie at predetermined intervals. This is because, to be able to draw any particular frame, the preceding key frame and all difference frames following that key frame (up to the frame to be drawn) must be processed. It would be prohibitively slow to jump to a random spot in a movie, or play a movie backwards, if it consisted of a single key frame and a bunch of difference frames. The maximum number of frames that can occur before a key frame is inserted is the *key frame rate*. A compressor may insert key frames more often than the specified key frame rate, however (for instance, at a scene change, where there is very little similarity between one frame and the following frame).

Note that spatial and temporal compression are not competing forms of compression. Indeed, most QuickTime movies employ *both* spatial and temporal compression, since the key frames of a movie are typically spatially compressed images. Note also that the use of temporal compression forces us to revise our understanding of the data stored in a QuickTime movie file. Up to now, we've tended to think of the movie data as a sequence of images. Now we see that it's more accurate to think of the movie data as a sequence of images (key frames) and changes to those images (difference frames). Only at playback time (that is, after the movie data is decompressed) do we get an actual series of images.

COMPRESSING IMAGES

The sample applications that we've developed so far in this series of articles can open both QuickTime movie files and image files and display them in windows on the screen. When the user selects the "Compress..." menu item in the Test menu, QTCompress executes this code:

```
case IDM_COMPRESS:
    if (QTFrame_IsImageWindow(myWindow))
        QTCmpr_CompressImage(myWindowObject);
    else
        QTCmpr_CompressSequence(myWindowObject);
    myIsHandled = true;
    break;
```

As you can see, if the frontmost window contains an image, then QTCompress calls the function `QTCmpr_CompressImage` (which we'll consider in this section); otherwise, it calls `QTCmpr_CompressSequence` (which we'll consider in the next section).

The `QTCmpr_CompressImage` function is built mainly around two routines provided by the standard image compression dialog component, `SCRequestImageSettings` and `SCCompressImage`. `SCRequestImageSettings` displays and manages the standard image compression dialog box (Figure 1), and `SCCompressImage` performs the actual compression of the image data into a new buffer. (As

Sponsored by:

Forbes

Microsoft®

IBM

**PIPER
MARSHALL
RUDNICK
& WOLFE LLP**

Silicon Valley Bank

THE STANDARD
ATTORNEYS AT LAW

Brobeck
ATTORNEYS AT LAW

Tough love for your startup.

Garage.com's
BOOTCAMP
FOR STARTUPS®

S. California February 6-7 **Chicago** March 27-28 **New York** May 7-8 **Silicon Valley** Fall 2001

Building a high technology or life sciences business? Attend Garage.com's Bootcamp for Startups conference. In two days of intensive, high-energy sessions, you'll get expert advice from major players on everything from fundraising and business development to recruiting and marketing. Find out how to refine your business model. Discover how to position—and pitch—your startup to investors. Network with peers and potential mentors. Find out more—and register online—at www.garage.com/bootcamp.

REGISTER TODAY

www.garage.com/bootcamp

garage.com™

Copyright ©2001 Garage.com. All rights reserved. Bootcamp for Startups is a registered trademark, and Garage.com is a trademark of Garage.com.

you've probably guessed, all functions provided by the standard image compression dialog component begin with the letters "SC".)

Getting the Image Pixel Map

The first thing we need to do when compressing an image is to draw it into an offscreen graphics world. We'll use the pixel map associated with that offscreen graphics world in two ways. First, we'll pass it to the `SCSetTestImagePixelFormat` function to set the thumbnail image in the image compression dialog box. Then, later on, we'll pass it to `SCCompressImage` as the source image to be compressed.

As we've seen in earlier articles, we can use graphics importers to open and draw image files. In fact, we already have an instance of a graphics importer component associated with the image file; it's the one we use to draw the image into the window on the screen (namely, `(**theWindowObject).fGraphicsImporter`). But here we're going to create a *new* graphics importer instance to draw the image into the offscreen graphics world. This is because we'll want to use the existing graphics importer to redraw the image in the on-screen window inside of the modal-dialog event filter procedure `QTCmpr_FilterProc` (defined later). It might in fact be possible to cleverly juggle the graphics importer's graphics world (using `GraphicsImportSetGWorld`), but I never managed to get that strategy to work properly. So let's create a new graphics importer instance for the image to be compressed:

```
myErr = GetGraphicsImporterFromFile(
    &(**theWindowObject).fFileFSSpec, &myImporter);
if (myErr != noErr)
    goto bail;

myErr = GraphicsImportGetNaturalBounds(myImporter, &myRect);
if (myErr != noErr)
    goto bail;
```

Now that we know the size of the image, we can use this code to create the requisite offscreen graphics world:

```
myErr = QTNewGWorld(&myImageWorld, 0, &myRect, NULL, NULL,
    kICMTempThenAppMemory);
if (myErr != noErr)
    goto bail;

// get the pixmap of the GWorld; we'll lock the pixmap, just to be safe
myPixelFormat = GetGWorldPixelFormat(myImageWorld);
if (!LockPixels(myPixelFormat))
    goto bail;
```

Finally, we need to draw the image into `myImageWorld`.

```
GraphicsImportSetGWorld(myImporter, (CGrafPtr)myImageWorld,
    NULL);
GraphicsImportDraw(myImporter);
```

At this point, the offscreen graphics world `myImageWorld` contains the image to be compressed.

Setting a Test Image

Now we want to display the standard image compression dialog box, to get the user's desired compression settings. To do this, we need to open an instance of the standard image compression dialog component, like so:

```
myComponent = OpenDefaultComponent(StandardCompressionType,
    StandardCompressionSubType);
```

Before we call the `SCRequestImageSettings` function to display the dialog box on the screen, we need to set the thumbnail picture (called the *test image*) that is displayed in the top-right part of the dialog box, by calling the `SCSetTestImagePixelFormat` function:

```
SCSetTestImagePixelFormat(myComponent, myPixelFormat, NULL,
    scPreferScaling);
```

Here, the first two parameters are the instance of the standard image compression dialog component and the pixel map that contains the image. The third parameter is a pointer to a `Rect` structure that specifies the *area of interest* in the pixel map that is to be used as the test image. Passing the value `NULL` means to use the entire pixel map as the test image, suitably reduced into the 80 pixel by 80 pixel area in the dialog box. The fourth parameter indicates how the image reduction is to occur; it can be any of these constants:

```
enum {
    scPreferCropping          = 1 << 0,
    scPreferScaling          = 1 << 1,
    scPreferScalingAndCropping = scPreferScaling |
                                scPreferCropping,
    scDontDetermineSettingsFromTestImage = 1 << 2
};
```

You can also pass the value 0 in the fourth parameter to request the component's default method of displaying the test image, which is currently a combination of scaling and cropping. Personally, I prefer just scaling the image to fit into the available space, so I've passed the value `scPreferScaling`.

Installing Extended Procedures

We're *almost* ready to call `SCRequestImageSettings` to display the dialog box. We need to do just one more thing to configure the dialog box, namely install one or more callback procedures that extend the basic functionality of the standard image compression dialog component. (These are therefore called *extended procedures*.) We install these callback procedures by calling the `SCSetInfo` function with the `scExtendedProcsType` selector:

```
SCSetInfo(theComponent, scExtendedProcsType, &gProcStruct);
```

Here, `gProcStruct` is an *extended procedures structure*, of type `SCEExtendedProcs`, which is defined like this:

```
struct SCEExtendedProcs {
    SCModalFilterUPP    filterProc;
    SCModalHookUPP      hookProc;
    long                refcon;
    Str31               customName;
};
```

The `filterProc` field specifies a *modal-dialog event filter function*, to handle events that are not handled by the standard image compression dialog component itself. As we've done in the past, we'll provide a filter function that looks for update events for our application's windows and redraws those windows accordingly. Listing 1 shows our custom filter function.

- Custom Destinations

- Electronic Transaction Processing

- Easy Trialware Creation

- Install or UnInstall

Get It Together With InstallerMaker™

- Installation From FTP or HTTP Sites

When Time Is Money, Get It Done Fast!

Build Smaller Installers - With StuffIt InstallerMaker, you'll build installers faster than ever before! Compress your installers an average of 15% smaller using the power of the StuffIt Engine®. Smaller installers download faster, provide added space on CDs and servers, and increase network bandwidth.

Quickly Create Demoware - Easily turn your application into a polished demo. Just set the number of days for the demo to be active, paste in the graphics, and you're done!

Archive Freshening - Automatically update your installer project file; eliminate repeated searches for modified files.

- Resource Installation

- Built-In Resource Compression

- ShrinkWrap Disk Image Support

Scripting Saves Time - InstallerMaker provides full scriptability for all features. Automating the build process saves time and helps ensure the integrity of your installer.

Trialware in Minutes - Creating trialware is a breeze with InstallerMaker. With just a few clicks, and no extra coding, you can create trialware that is e-commerce ready.

- Marketing Opportunities To Help You Make Money

Update in a Flash - Easily build intelligent "diff" files in installers to create small updaters for quick online distribution. From one simple installer, you can update 68k, PowerPC or FAT applications.

- Hierarchical Package Support

More Details Online - There's a lot more InstallerMaker can do for you. Get all the info at www.aladdinsys.com.

- Built-In Updaters



Listing 1: Filtering events in the image compression dialog box

```

QTCmpr_FilterProc
static PASCAL_RTN Boolean QTCmpr_FilterProc
(DialogPtr theDialog, EventRecord *theEvent,
short *theItemHit, long theRefCon)
{
#pragma unused(theItemHit, theRefCon)
Boolean myEventHandled = false;
WindowRef myEventWindow = NULL;
WindowRef myDialogWindow = NULL;

#if TARGET_API_MAC_CARBON
myDialogWindow = GetDialogWindow(theDialog);
#else
myDialogWindow = theDialog;
#endif

switch (theEvent->what) {
case updateEvt:
// update the specified window, if it's behind the modal dialog box
myEventWindow = (WindowRef)theEvent->message;
if ((myEventWindow != NULL) &&
(myEventWindow != myDialogWindow)) {
#if TARGET_OS_MAC
QTFramE_HandleEvent(theEvent);
#endif
myEventHandled = false;
}
break;
}

return(myEventHandled);
}

```

This is a fairly typical event filter function. It looks for update events that are not destined for the dialog box and sends them to the framework's event-handling function. Notice that this step isn't necessary on Windows; on that platform, redraw messages are sent directly to the window procedure of the affected window.

We can also install a *hook function*, which is called whenever the user selects (or "hits") an item in the dialog box. We can then intercept those hits and handle them in any way we like. A typical way to use the hook function is in connection with a custom button in the standard image compression dialog box. **Figure 6** shows the dialog box with a new button labeled "Defaults".

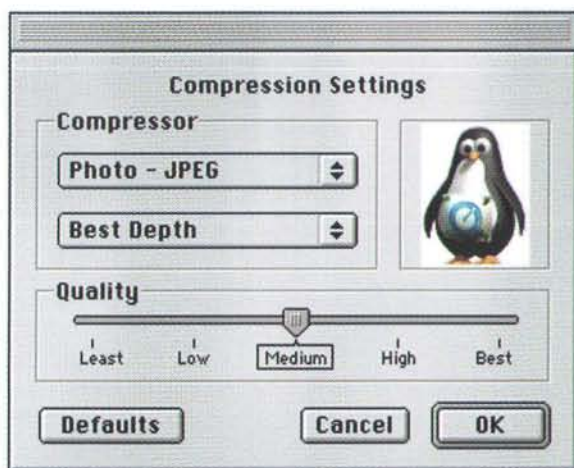


Figure 6: The standard image compression dialog box with a custom button.

We install this custom button by specifying a name for the button in the `customName` field of the extended functions structure. We can do that with these two lines of code:

```

StringPtr myButtonTitle =
QTUtils_ConvertCToPascalString("Defaults");

BlockMove(myButtonTitle, gProcStruct.customName,
myButtonTitle[0] + 1);

```

Then we can handle user clicks on this custom button inside our hook function, `QTCmpr_ButtonProc`, defined in Listing 2.

Listing 2: Intercepting events in a hook function

```

QTCmpr_ButtonProc
static PASCAL_RTN short QTCmpr_ButtonProc
(DialogPtr theDialog, short theItemHit,
void *theParams, long theRefCon)
{
#pragma unused(theDialog)
// in this sample code, we'll have the settings revert to their default values
// when the user clicks on the custom button
if (theItemHit == scCustomItem)
SCDefaultPixMapSettings(theParams,
(PixMapHandle)theRefCon, false);

// always return the item passed in
return(theItemHit);
}

```

This hook function is extremely simple; it just looks for hits on the custom button (signaled by the constant `scCustomItem`) and then calls the `SCDefaultPixMapSettings` function to reset the dialog box to its default values. Notice that the reference constant passed to our hook function (in the `theRefCon` parameter) is expected to be a handle to the pixel map we created earlier. We install this reference constant by setting the `refcon` field of the extended procedures structure. Listing 3 shows our definition of the `QTCmpr_InstallExtendedProcs` function, which we use to set up our extended procedures.

Listing 3: Installing the extended procedures

```

QTCmpr_InstallExtendedProcs
static void QTCmpr_InstallExtendedProcs
(ComponentInstance theComponent, long theRefCon)
{
StringPtr myButtonTitle =
QTUtils_ConvertCToPascalString(kButtonTitle);

// the modal-dialog filter function can be used to handle any events that
// the standard image compression dialog handler doesn't know about, such
// as any update events for windows owned by the application
gProcStruct.filterProc =
NewSCModalFilterUPP(QTCmpr_FilterProc);

#if USE_CUSTOM_BUTTON
// the hook function can be used to handle clicks on the custom button
gProcStruct.hookProc =
NewSCModalHookUPP(QTCmpr_ButtonProc);

// copy the string for our custom button into the extended procs structure
BlockMove(myButtonTitle, gProcStruct.customName,
myButtonTitle[0] + 1);
#else
gProcStruct.hookProc = NULL;
gProcStruct.customName[0] = 0;
#endif

// in this example, we pass the pixel map handle as a refcon
gProcStruct.refcon = theRefCon;

// set the current extended procs
SCSetInfo(theComponent, scExtendedProcsType, &gProcStruct);

free(myButtonTitle);
}

```


You'll notice that we've used the compiler flag `USE_CUSTOM_BUTTON` to indicate whether we want to install a custom button in the standard image compression dialog box. Some image compressors want to install an Options button in that dialog box, and our custom button would prevent them from doing so. (See **Figure 7**, later, for a dialog box that contains an Options button.) For this reason, we usually won't install a custom button. But you should at least know how to do so.

Finally we can call `QTCmpr_InstallExtendedProcs` to install our extended procedures and then `SCRequestImageSettings` to display the dialog box.

```
if (gUseExtendedProcs)
    QTCmpr_InstallExtendedProcs(myComponent, (long)myPixMap);

myErr = SCRequestImageSettings(myComponent);
```

Compressing the Image

If the user selects the Cancel button in the standard image compression dialog box, then `SCRequestImageSettings` returns the value `scUserCancelled`. Otherwise, if `SCRequestImageSettings` returns `noErr`, we want to go ahead and compress the image. Thankfully, we can do this with a single call to the function `SCCompressImage`.

```
myErr = SCCompressImage(myComponent, myPixMap, NULL,
                        &myDesc, &myHandle);
```

`SCCompressImage` compresses the specified pixel map using the current settings of the specified standard image compression dialog component. It allocates storage for the compressed image and returns a handle to that storage in the fifth parameter (`myHandle`). It also returns an image description in the fourth parameter (`myDesc`). We can write the compressed data into a new file by calling the application function

`QTCmpr_PromptUserForDiskFileAndSaveCompressed`, passing in the compressed data and the image description. (See the file `QTCompress.c` in this month's source code for the definition of this function.) Listing 4 shows our complete definition of `QTCmpr_CompressImage`.

Listing 4: Compressing an image

```
QTCmpr_CompressImage
void QTCmpr_CompressImage (WindowObject theWindowObject)
{
    Rect myRect;
    GraphicsImportComponent myImporter = NULL;
    ComponentInstance myComponent = NULL;
    GWorldPtr myImageWorld = NULL;
    PixMapHandle myPixMap = NULL;
    ImageDescriptionHandle myDesc = NULL;
    Handle myHandle = NULL;
    OSErr myErr = noErr;

    if (theWindowObject == NULL)
        return;

    myErr = GetGraphicsImporterForFile(
        &(**theWindowObject).fFileFSSpec, &myImporter);
    if (myErr != noErr)
        goto bail;

    myErr = GraphicsImportGetNaturalBounds(myImporter,
        &myRect);
    if (myErr != noErr)
        goto bail;

    // create an offscreen graphics world and draw the image into it
    myErr = QTNewGWorld(&myImageWorld, 0, &myRect, NULL, NULL,
        kICMTempThenAppMemory);
    if (myErr != noErr)
        goto bail;

    // get the pixmap of the GWorld; we'll lock the pixmap, just to be safe
    myPixMap = GetGWorldPixMap(myImageWorld);
    if (!LockPixels(myPixMap))
        goto bail;

    // set the current port and draw the image
    GraphicsImportSetGWorld(myImporter, (CGrafPtr)myImageWorld,
```

“Listen Different”
www.macshowlive.com

Every week, we're
 streaming live,
 interactive,
 and talking...

100% Mac!

Shawn King

Hilary Percy

• News • Basics • Gaming • Mac Pro • Tech Tips • Feature Interviews

listen anytime @ macshowlive.com

MAC LIVE SHOW


```

        NULL);
GraphicsImportDraw(myImporter);

// open the standard compression dialog component
myComponent = OpenDefaultComponent(StandardCompressionType,
                                   StandardCompressionSubType);
if (myComponent == NULL)
    goto bail;

// set the picture to be displayed in the dialog box
SCSetTestImagePixmap(myComponent, myPixmap,
                     NULL, scPreferScaling);

// install the custom procs, if requested
if (gUseExtendedProcs)
    QTCmpr_InstallExtendedProcs(myComponent, (long)myPixmap);

// request image compression settings from the user
myErr = SCRequestImageSettings(myComponent);
if (myErr == scUserCancelled)
    goto bail;

// compress the image
myErr = SCCompressImage(myComponent, myPixmap, NULL,
                       &myDesc, &myHandle);
if (myErr != noErr)
    goto bail;

// save the compressed image in a new file
QTCmpr_PromptUserForDiskFileAndSaveCompressed(myHandle,
                                              myDesc);

bail:
    if (gUseExtendedProcs)
        QTCmpr_RemoveExtendedProcs();

    if (myPixmap != NULL)
        if (GetPixelsState(myPixmap) & pixelsLocked)
            UnlockPixels(myPixmap);

    if (myImporter != NULL)
        CloseComponent(myImporter);

    if (myComponent != NULL)
        CloseComponent(myComponent);

    if (myDesc != NULL)
        DisposeHandle((Handle)myDesc);

    if (myHandle != NULL)
        DisposeHandle(myHandle);

    if (myImageWorld != NULL)
        DisposeGWorld(myImageWorld);
}

```

COMPRESSING IMAGE SEQUENCES

Compressing a sequence of images is not fundamentally different from compressing a single image. We'll need to display the standard image compression dialog box, as before, to get the user's desired compression settings. Then, however, instead of compressing a single image, we'll need to loop through all the images in the sequence and compress each one individually. We'll obtain our sequence of images by extracting the individual frames from an existing QuickTime movie, and then we'll write the compressed images into a new QuickTime movie. (In effect, we'll be converting a QuickTime movie from one compression scheme to another; this operation is often called *transcoding*.) So we'll have the added overhead of creating a new QuickTime movie, track, and media, and of adding samples to the media by calling `AddMediaSample`. We've done this kind of thing numerous times before, so that part of the code shouldn't slow us down too much.

When we want to compress a sequence of images, we need to call `SCRequestSequenceSettings` instead of `SCRequestSettings`. The

dialog box that it displays is shown in **Figure 7**; as you can see, it contains an additional pane of controls for specifying the number of frames per second, the key frame rate, and the maximum *data rate* (the number of bytes per second that can be processed).

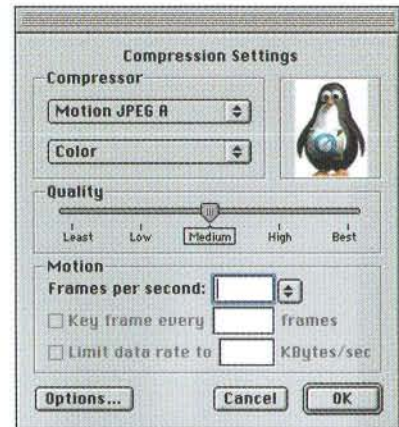


Figure 7: The standard image compression dialog box for an image sequence.

When configuring the dialog box and when responding to its dismissal, we'll need to add code to handle this additional information. So let's get started.

Getting the Image Sequence

As mentioned above, we're going to obtain our sequence of images by reading the individual frames from the video track of a QuickTime movie. (Let's call this the *source movie*.) We can get the source movie by reading the `fMovie` field of the window object record, and we can get the source movie's video track by calling the `GetMovieIndTrackType` function:

```

mySrcMovie = (**theWindowObject).fMovie;
if (mySrcMovie == NULL)
    goto bail;

mySrcTrack = GetMovieIndTrackType(mySrcMovie, 1,
                                  VideoMediaType, movieTrackMediaType);
if (mySrcTrack == NULL)
    goto bail;

```

To make things look a bit cleaner, we don't want the movie to be playing while we are compressing its frames into a new movie, so we call `SetMovieRate` to stop the movie. We also need to keep track of the current movie time, since we'll be changing it as we move from frame to frame through the movie. Later, when we're done recompressing the frames of the movie, we'll reset the movie time to this saved value.

```

SetMovieRate(mySrcMovie, (Fixed)0L);
myOrigMovieTime = GetMovieTime(mySrcMovie, NULL);

```

Finally, we need to know how many video frames are in the source movie, so that we know (for instance) how many iterations our loop should make. `QTCmpr_CompressSequence` includes this line of code for counting the frames of the source movie:

```

myNumFrames = QTUtils_GetFrameCount(mySrcTrack);

```


"I registered my practice"

www.thecattherapist.com



Carole Wilbourn, world renown Cat Therapist, registered her domain name to put her business online at register.com. Visit us at www.register.com or call us at 1-800-7-WWW-NET, and we'll help you register your domain name right over the phone.

register
comTM
the *first* step on the webSM

There are several methods we could use to determine how many frames the source movie contains. Probably the best method is just to step through the interesting times in the movie using the `GetTrackNextInterestingTime` function, as shown in Listing 5.

Listing 5: Counting the frames in a movie

```

QTUtils_GetFrameCount
long QTUtils_GetFrameCount (Track theTrack)
{
    long    myCount = -1;
    short   myFlags;
    TimeValue myTime = 0;

    if (theTrack == NULL)
        goto bail;

    // we want to begin with the first frame (sample) in the track
    myFlags = nextTimeMediaSample + nextTimeEdgeOK;

    while (myTime >= 0) {
        myCount++;

        // look for the next frame in the track; when there are no more frames,
        // myTime is set to -1, so we'll exit the while loop
        GetTrackNextInterestingTime(theTrack, myFlags, myTime,
                                   fixed1, &myTime, NULL);

        // after the first interesting time, don't include the time we're currently at
        myFlags = nextTimeStep;
    }

bail:
    return(myCount);
}

```

For more discussion of `GetTrackNextInterestingTime`, see “Word Is Out” in *MacTech*, November 2000.

Configuring the Standard Image Compression Dialog Component

As before, we need to open an instance of the standard image compression dialog component and configure the initial settings of the dialog box. Opening an instance of the component uses the same code we used in the case of a single image:

```

myComponent = OpenDefaultComponent(StandardCompressionType,
StandardCompressionSubType);

```

To configure the settings in the dialog box, we first want to turn off the “Best Depth” menu option in the pixel depth pop-up menu. This is because we are going to draw the movie frames into a 32-bit offscreen graphics world, regardless of the pixel depth of the original source images. A better approach might be to determine the maximum bit depth used in the source images (by looping through the video sample descriptions of the video frames) and then create an offscreen graphics world of that depth. (This refinement, of course, is left as an exercise for the reader.) We can disable the “Best Depth” option using this code:

```

SCGetInfo(myComponent, scPreferenceFlagsType, &myFlags);
myFlags &= ~scShowBestDepth;
SCSetInfo(myComponent, scPreferenceFlagsType, &myFlags);

```

Next, we want to allow the user to leave the frame rate field blank (in which case the compression component will preserve the original frame durations of the source movie). To do this, we need to specify that 0 is an acceptable value in that field. We do that by executing these lines of code:

```

SCGetInfo(myComponent, scPreferenceFlagsType, &myFlags);
myFlags |= scAllowZeroFrameRate;
SCSetInfo(myComponent, scPreferenceFlagsType, &myFlags);

```

If the user enters a number in the frame rate field, we'll use that number as the new sample rate for the destination movie.

Setting the Test Image

Before we display the compression settings dialog box to the user, we want to set the test image. In the present case, however, we have an entire sequence of images to handle, not just a single image. Which of those images shall we select as the test image? Let's select the movie poster image, on the assumption that that image is representative of the content of the entire sequence of images (that is, of the source movie itself). So we can call `GetMoviePosterPict` to get a `PicHandle` to the test image:

```

myPicture = GetMoviePosterPict(mySrcMovie);

```

Then we can get the size of the poster image and create an offscreen graphics world large enough to hold that image:

```

GetMovieBox(mySrcMovie, &myRect);
myErr = NewGWorld(&myImageWorld, 32, &myRect, NULL, NULL,
                 0L);

```

And, as before, we'll lock the pixel map of that graphics world:

```

myPixMap = GetGWorldPixMap(myImageWorld);
if (!LockPixels(myPixMap))
    goto bail;

```

Next we want to draw the poster image into the offscreen graphics world. Since we've got a handle to a QuickDraw picture, we can use the `DrawPicture` function to draw the picture. First, however, we need to make sure to set the current graphics world to our new offscreen graphics world and to erase the destination graphics world.

```

GetGWorld(&mySavedPort, &mySavedDevice);
SetGWorld(myImageWorld, NULL);
EraseRect(&myRect);
DrawPicture(myPicture, &myRect);
KillPicture(myPicture);
SetGWorld(mySavedPort, mySavedDevice);

```

Finally we are ready to call `SCSetTestImagePixMap` to set the test image:

```

SCSetTestImagePixMap(myComponent, myPixMap, NULL,
                    scPreferScaling);

```

Displaying the Compression Settings Dialog Box

Once again, we have a couple of things still to do before we can display the standard image compression dialog box. For one thing, we need to install the extended procedures; here we can use exactly the same application function as in the single-image case:

```

if (gUseExtendedProcs)
    QTCmpr_InstallExtendedProcs(myComponent, (long)myPixMap);

```

Next, we want to set some default settings for the dialog box. The standard image compression dialog component can examine the pixel map that we just created and derive some sensible default

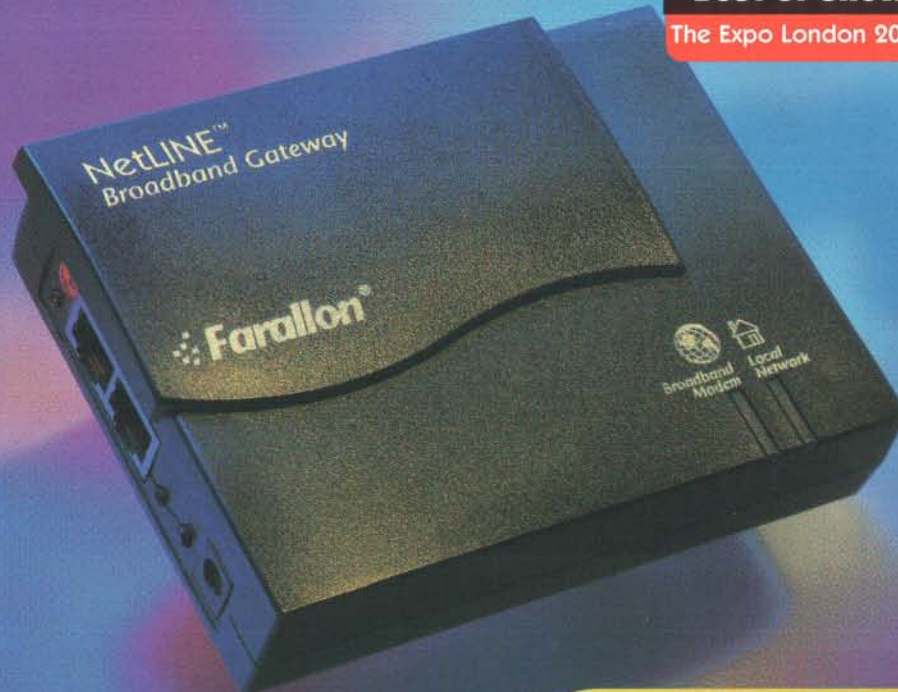
Share your high-speed Internet connection and protect your data from unauthorized access with the

NetLINE™ Broadband Gateway

Farallon
www.farallon.com

Best of Show

The Expo London 2000



The perfect solution for home, office or school!

The NetLINE Broadband Gateway

allows you to share a high-speed Internet connection, such as Cable or DSL, with multiple computers using just a single IP address from your ISP.

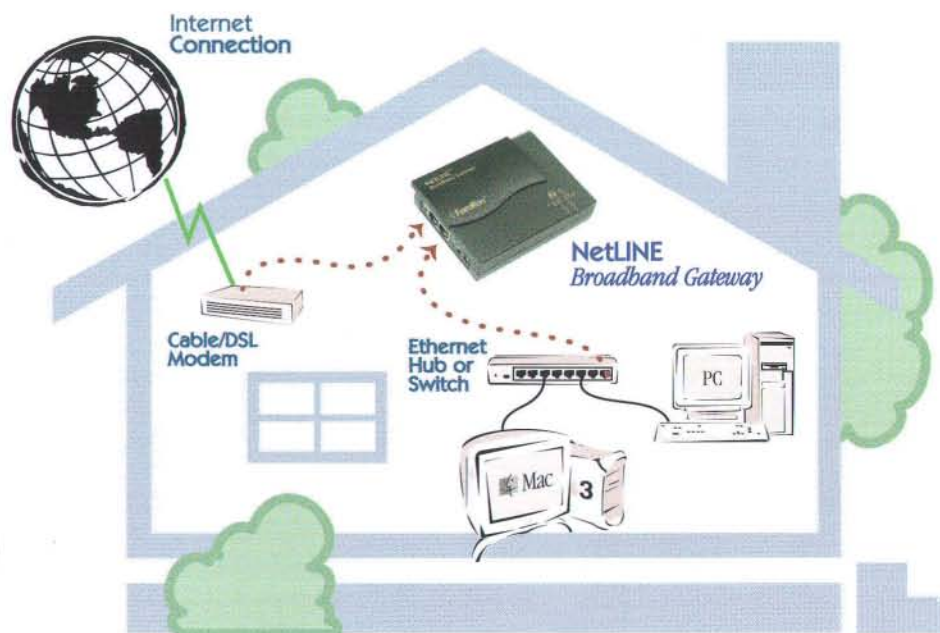
The built-in firewall will protect your important files by preventing unauthorized access via the Internet.

Works with both Macs and PCs. Web-based configuration makes set-up a breeze.

Receive a coupon for 25% off a Farallon hub or switch when you purchase the NetLINE Broadband Gateway. (US Only)



For more information contact Dr. Farallon at 1-800-613-4954 or visit us on the web at **www.farallon.com**



The NetLINE Broadband Gateway connects between your network and your Cable/DSL modem

Available at **4@DEPOT**



- no complicated reseller rebates
- no special terms or conditions
- no hidden charges
- no "get this if you buy this"
- no hooks, catches, installation fees, or well, any other bull!

OS X

Only \$89 at PowerMax!



- Aqua user interface
- Super-modern core operating system
- Killer graphics
- Gentle migration
- Internet savvy
- Call PowerMax for expert advice & info!



Authorized Reseller



Best price ever on a Cube! only: \$1199*

Graphite iBook™ G3/466 64/10/DVD \$1499*

G4/500 w/256 RAM, 27 Gb HD, DVD-ROM, Zip \$2149*

Great deals on used & refurbished Mac! Models, fully configured..... Starting At

Pre-PowerPC Macs.....	\$29
6100 series.....	\$99
7200 series.....	\$149
PowerBase Desktops.....	\$219
PowerCenter Pro Towers.....	\$249
6400 series.....	\$219
7500 series.....	\$249
7600 series.....	\$279
7300 series.....	\$299
8500 series.....	\$339
Umax S900 Towers w/6 PCI.....	\$389
PowerBook 5300C series.....	\$459
9500 series.....	\$488
8600 series.....	\$499
9600 series.....	\$699
iMac™ G3/233 Bondi Blue.....	\$569
Beige G3 Desktops.....	\$639
iMac™ G3/266 various colors.....	\$588
iMac™ G3/350 Blueberry.....	\$639
Beige G3 Towers.....	\$679
PowerBook™ 1400C series.....	\$699
PowerBook G3's.....	\$899
Blue & White G3 Towers.....	\$949
iBooks™.....	\$999
G4 Graphite Towers.....	\$999
Factory & Reseller Refurbished Macs™	
iMac™ G3/350 64/6/24x CD.....	\$649
iMac™ G3/400 64/10/DVD.....	\$798
Graphite iMac™ G3/400 128/13.....	\$888
Blue & Wh. G3/400 64/20.....	\$1088
Cube G4/450 64/20/DVD.....	\$1199
Graphite iBook 64/10/DVD.....	\$1499
PowerBook® 400 64/6/DVD.....	\$1799
G4/500 256/27/DVD/Zip.....	\$2149
G4/500 DP 256/40/DVD-RAM.....	\$2399

Regular price\$129
Less PowerMax instant discount-\$10
Less instant in-store credit toward the purchase of any 128 MB RAM upgrade or better.....-\$30
Net cost to you.....\$89

Cube & display w/extra RAM
 G4/450 Cube* upgraded to 128 mb RAM, 20 Gb drive, DVD-ROM drive and a 17" Apple Studio display.* Extend your warranty to three years for only \$249.



\$1549

The "Beige is Beautiful" Bundle
 Get this rock-solid G3/266 Tower complete with 128 mb RAM, 6 Gb hard drive, internal Zip drive and beautiful Apple 1710 Trinitron display for one great low price!



\$999

iMac™ w/extra RAM & printer
 Graphite S.E. G3/400 iMac™ DV w/ 192 mb RAM, 13 Gb drive, DVD-ROM and a Canon BJC-2100 printer. Extend your warranty to 3 years for only \$149. Price after Canon mfrg. rebate.



\$849

G4 & 17" display w/extra RAM
 New G4/400 Tower upgraded to 128 mb RAM, 20 Gb drive, DVD-ROM drive and a 17" Apple Studio display.* Extend your warranty to three years for only \$249.



\$1699

TradeUp
 We'll take your Mac OS® computer in trade toward the purchase of new product. Call one of our expert Mac consultants for full details!

800-689-8198
www.powermax.com

Local: (503) 624-1827 • Fax (503) 624-1635
email: sales@powermax.com

- Personal Financing • Fast P.O. Approvals • Business Leasing
- Weekly specials on our web site • Order on line or call one of our experts!

POWER MAX



Apple Specialist

Prices subject to change without notice. Prices reflect cash discount. Credit card orders strictly verified against fraudulent use. With use of credit card as payment customer acknowledges that some products are subject to final sale. Many prices are limited to stock on hand. All brand or product names are registered trademarks of their respective holders.

Knowledge is Power



settings based on the characteristics of that image. So let's take advantage of that capability:

```
SCDefaultPixMapSettings(myComponent, myPixMap, true);
```

Also, we want to clear out whatever default frame rate was selected by the standard image compression dialog component. As we discussed above, we'd like to use the frame rate 0, indicating that the frame rate of the source movie should be used. (The user is free to change this rate, but at least we want the default value to be 0.) We can first retrieve and then reset the current temporal settings of the component:

```
myErr = SCGetInfo(myComponent, scTemporalSettingsType,
                  &myTimeSettings);
if (myErr != noErr)
    goto bail;
```

```
myTimeSettings.frameRate = 0;
SCSetInfo(myComponent, scTemporalSettingsType,
          &myTimeSettings);
```

SCGetInfo and SCSetInfo expect temporal settings to be stored in a structure of type SCTemporalSettings, defined like this:

```
struct SCTemporalSettings {
    CodecQ          temporalQuality;
    Fixed           frameRate;
    long            keyFrameRate;
};
```

Finally, we're ready to call SCRequestSequenceSettings to display the standard image compression dialog box:

```
myErr = SCRequestSequenceSettings(myComponent);
if (myErr == scUserCancelled)
    goto bail;
```

Adjusting the Sample Count

If we've made it this far in the QTCmpr_CompressSequence function, we have successfully displayed the standard image compression dialog box and the user has selected his or her desired compression settings. In the single-image case, we could finish up rather quickly, by immediately calling SCCompressImage and then saving the compressed data into a new file. In the current case, however, we still have a good bit of work left to do. We need to retrieve the temporal settings — which may indicate a new frame rate for the destination movie — and configure the destination movie accordingly. Then we need to step through the frames of the source movie and compress each frame in the movie.

We'll begin by first retrieving the temporal settings selected by the user:

```
myErr = SCGetInfo(myComponent, scTemporalSettingsType,
                  &myTimeSettings);
```

If the user wants to change the frame rate of the movie (as indicated by a non-zero value in the frameRate field of the temporal setting structure myTimeSettings), then we need to calculate the number of frames in the destination movie and the duration of the destination movie. We can do that like this:

```
if (myTimeSettings.frameRate != 0) {
    long myDuration = GetMovieDuration(mySrcMovie);
    long myTimeScale = GetMovieTimeScale(mySrcMovie);
```



```
float myFloat = (float)myDuration *
                myTimeSettings.frameRate;

myNumFrames = myFloat / myTimeScale / 65536;
if (myNumFrames == 0)
    myNumFrames = 1;
}
```

Creating the Target Movie

Suppose now that `myFile` is a file system specification for the destination movie file (perhaps we called our framework function `QTFrame_PutFile` to elicit that file specification from the user). At this point, we need to create the destination movie file and movie, as shown in Listing 6.

Listing 6: Creating a new movie file and movie

```
myErr = CreateMovieFile(&myFile, sigMoviePlayer,
                      QTCmpr_CompressSequence,
                      smSystemScript,
                      createMovieFileDeleteCurFile |
                      createMovieFileDontCreateResFile,
                      &myRefNum, &myDstMovie);

if (myErr != noErr)
    goto bail;

// create a new video movie track with the same dimensions as the entire source movie
myDstTrack = NewMovieTrack(myDstMovie,
                          (long)(myRect.right - myRect.left) << 16,
                          (long)(myRect.bottom - myRect.top) << 16,
                          kNoVolume);

if (myDstTrack == NULL)
    goto bail;

// create a media for the new track with the same time scale as the source movie;
// because the time scales are the same, we don't have to do any time scale conversions
myDstMedia = NewTrackMedia(myDstTrack, VIDEO_TYPE,
                          GetMovieTimeScale(mySrcMovie), 0, 0);

if (myDstMedia == NULL)
    goto bail;
```

We of course want to copy the movie user data and other settings from the source movie to the destination movie:

```
CopyMovieSettings(mySrcMovie, myDstMovie);
```

Also, we want to set the movie matrix of the destination movie to the identity matrix and clear out the movie clip region. This is because the process of recompressing the movie transforms and composites all the video tracks into a single, untransformed video track.

```
SetIdentityMatrix(&myMatrix);
SetMovieMatrix(myDstMovie, &myMatrix);
SetMovieClipRgn(myDstMovie, NULL);
```

Finally, since we're about to start adding compressed video samples to the destination movie, we need to call `BeginMediaEdits`:

```
myErr = BeginMediaEdits(myDstMedia);
if (myErr != noErr)
    goto bail;
```

Compressing the Image Sequence

We are now ready to start compressing the frames of the source movie into the destination movie. We are going to step through the source movie, draw the current frame into our offscreen graphics world, compress that data, and then add the compressed data as a video sample to the destination movie. To start things off, we call `SCCompressSequenceBegin`:

```
myErr = SCCompressSequenceBegin(myComponent, myPixMap, NULL,
                                &myImageDesc);
```

Software Protection

WIBU-KEY

USB and ADB for Macintosh

WIBU-KEY is the most advanced Mac copy protection & license management solution available today. The 3.0 Mac driver is available for both MacOS 9 and MacOS X.



The USB hardware is the same for the Macintosh and the PC. Inventory one type of key and deliver it with Macintosh and Windows versions of your software. You can even manage network licenses across heterogeneous networks.

WIBU-SYSTEMS has been a member of USB Implementers Forum since 1997. As the first copy protection product certified by the USB-IF, the WIBU-BOX/U successfully completed all USB Compliance tests and is officially included in the USB Integrators List.



Best Service
New Technologies
High Quality Products

Order Your Test Kit
1-800-986-6578
www.wibu.com



USA, Canada: Griffin Technologies, LLC
916 Massachusetts Street, Lawrence, KS 66044
phone: (785) 832-2070 • fax: (785) 832-8787
email: sales@griffintech.com • www.griffintech.com

Argentina, Australia, Belgium,
Canada, Croatia, Denmark, Finland,
France, Germany, Great Britain, Hungary,
Japan, Jordan, Korea, Lebanon, Netherlands,
Spain, Portugal, Syria, USA

International: WIBU-SYSTEMS AG
Rueppurrer Strasse 52-54 • D-76137 Karlsruhe
phone: +49-721-93172-0 • fax: +49-721-93172-22
email: info@wibu.com • http://www.wibu.com



Quality the World Trusts

SCCompressSequenceBegin initiates a compression sequence, using the current settings of the specified component instance and the characteristics of the specified pixel map. The third parameter is a pointer to a `Rect` structure that indicates what portion of the pixel map we're going to be compressing; the value `NULL` indicates that we want to compress the entire pixel map. The fourth parameter is a pointer to an image description. The standard image compression dialog component will allocate storage for that description and fill in its fields when it compresses some data for us. In turn, we shall use that image description when we call `AddMediaSample`.

Next we need to erase the offscreen graphics world that we're going to be drawing movie frames into and set the source movie to draw its frames into that graphics world:

```
SetGWorld(myImageWorld, NULL);
EraseRect(&myRect);
SetMovieGWorld(mySrcMovie, myImageWorld,
    GetGWorldDevice(myImageWorld));
```

From now on, until we reset the movie graphics world, calling `MoviesTask` on the source movie will cause the current movie frame to be drawn into `myImageWorld`.

So let's start looping through the frames of the source movie. We begin by setting a variable that holds the current movie time to the beginning of the movie and by retrieving the duration of the source movie:

```
myCurMovieTime = 0;
mySrcMovieDuration = GetMovieDuration(mySrcMovie);
```

The remainder of our code in this section will occur inside of a for loop that iterates through all the frames of the source movie:

```
for (myFrameNum = 0; myFrameNum < myNumFrames; myFrameNum++)
{
    // get a frame, compress it, and add it to the new movie
}
```

The first thing we want to do inside this loop is get the next frame of the source movie. If we are not resampling the movie (that is, changing the frame rate), we can use `GetMovieNextInterestingTime` to step us forward in the movie; otherwise, we need to step forward by the appropriate amount, based on the desired new frame rate. Listing 7 shows our code for this step.

Listing 7: Getting the next frame of the source movie

QTCompr_CompressSequence

```
if (myTimeSettings.frameRate) {
    myCurMovieTime = myFrameNum * mySrcMovieDuration /
        (myNumFrames - 1);
    myDuration = mySrcMovieDuration / myNumFrames;
} else {
    OSType myMediaType = VIDEO_TYPE;

    myFlags = nextTimeMediaSample;

    // if this is the first frame, include the frame we are currently on
    if (myFrameNum == 0)
        myFlags |= nextTimeEdgeOK;

    // if we are maintaining the frame durations of the source movie,
    // skip to the next interesting time and get the duration for that frame
    GetMovieNextInterestingTime(mySrcMovie, myFlags, 1,
        &myMediaType, myCurMovieTime, 0, &myCurMovieTime,
        &myDuration);
}
```

Then we need to set the current movie time of the source movie to the time just calculated and draw the movie into the offscreen graphics world:

```
SetMovieTimeValue(mySrcMovie, myCurMovieTime);
MoviesTask(mySrcMovie, 0);
MoviesTask(mySrcMovie, 0);
MoviesTask(mySrcMovie, 0);
```

(Why *three* calls to `MoviesTask` here? Partly it's to make sure that the current movie frame is completely drawn before continuing with our work, and partly it's just because that's the way the sample code on which `QTCompr_CompressSequence` is based was written. Consider this your first taste of QuickTime black magic. No harm results from a few extra calls here.)

If the user has requested that the data rate of the destination movie be constrained, then we need to tell the standard image compression dialog component the duration of the current frame, in milliseconds. We can do that using this code:

```
if (!SCGetInfo(myComponent, scDataRateSettingsType,
    &myRateSettings)) {
    myRateSettings.frameDuration = myDuration * 1000 /
        GetMovieTimeScale(mySrcMovie);
    SCSetInfo(myComponent, scDataRateSettingsType,
        &myRateSettings);
}
```

Finally we can actually compress the pixel map data, by calling `SCCompressSequenceFrame`:

```
myErr = SCCompressSequenceFrame(myComponent, myPixMap,
    &myRect, &myCompressedData, &myDataSize, &mySyncFlag);
```

MacWise

The best in Macintosh
terminal emulation

Version 4.26

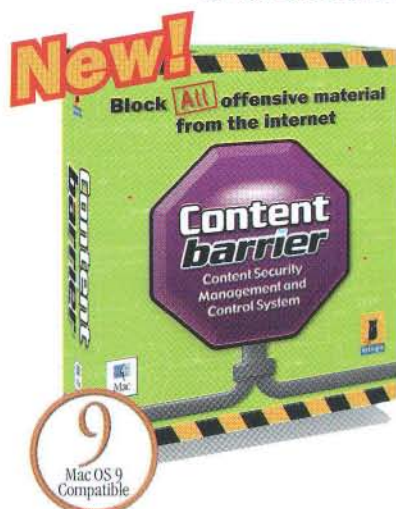
Emulations Include
Wyse 50, 60, and 370
Viewpoint, VT100, Esprit, TV925
Color and Fkeys supported
Carnation
Software
512.858.9234

www.carnation-software.com

You think your kids are safe on the Net. Think again...



ContentBarrier. The new Parental Control solution for your Macintosh.



You don't let your kids read just anything, do you? Well, when they surf the Internet, they can see whatever they want, unless you're there to watch over their shoulders. With ContentBarrier, the Internet is a safer place for your children.

ContentBarrier sets up a protective wall around your computer. Its pre-defined categories let you choose what you don't want your children to see, and you can also create your own custom categories. Inappropriate web sites are blocked, shielding your children from things they are too young for.

ContentBarrier blocks adult web sites, sites with subjects not fit for children, and even blocks chats when predatory language is used. It blocks all offensive content coming from the Internet.

ContentBarrier works with multiple users; if you have several children, you can set different criteria, corresponding to their age or maturity. You can choose whether they have access to newsgroups, e-mail, or whether they can download files. You can set the program to let them only use the Internet at certain times, and on specific days. The program can even send you e-mail, automatically, when certain events occur.

ContentBarrier keeps a complete log of all web sites visited, whether blocked or not. This allows you to have a full record of exactly what your children are doing on the Internet. You can also take an inventory of your children's computer, and find all pictures, movies, music files and web pages.

ContentBarrier makes the Internet a safer place for your children.

Licences available
5, 10, 20+

www.intego.com



ContentBarrier is fully compatible with Intego's acclaimed NetBarrier and VirusBarrier

CDW
800-509-4239

New sold at
ClubMac
www.clubmac.com
800-258-2622

@DEPOT
800-622-3381

J&R
800-221-8180

MacMail
800-217-9492

MacNN
★★★★★

APPLE WIZARDS
★★★★★

MacWarehouse
800-255-6227

macZone
800-436-0606

Outpost.com
877-688-7678

COMPUSA

MICRO CENTER
www.microcenter.com

buyonet
www.buyonet.com

Macworld
★★★★★

MACGENERATION



Mac and the Mac logo are trademarks of Apple Computer, Inc., registered in the U.S. and other countries. Outpost.com™ is a trademark of Cyberian Outpost, Inc. All other trademarks belong to their respective owners. Illustration Mona Daly ©2000 Artville Stock Images.

If `SCCompressSequenceFrame` completes successfully, then `myCompressedData` will hold a handle to the compressed data and `myDataSize` will be the size of the compressed data. In addition, the `mySyncFlag` parameter will hold a value that indicates whether the compressed frame is a key frame (0) or a difference frame (`mediaSampleNotSync`). We will pass this value directly to the `AddMediaSample` function, like this:

```
myErr = AddMediaSample(myDstMedia, myCompressedData, 0,
                      myDataSize, myDuration,
                      (SampleDescriptionHandle)myImageDesc,
                      1, mySyncFlag, NULL);
```

And so we've completely handled a frame of the source movie.

Finishing Up

Once we've exited the for loop that iterates through all the frames of the source movie, we need to do a little cleaning up. First, we need to close the compression sequence that we opened by calling `SCCompressSequenceBegin`:

```
SCCompressSequenceEnd(myComponent);
```

`SCCompressSequenceEnd` disposes of the image description and compressed data handles allocated by `SCCompressSequenceBegin`.

Next we need to close our media-editing session by calling `EndMediaEdits`:

```
myErr = EndMediaEdits(myDstMedia);
```

Then we proceed as normal, inserting the edited media into the track and adding the movie resource to the destination movie file.

```
InsertMediaIntoTrack(myDstTrack, 0, 0,
                    GetMediaDuration(myDstMedia), fixed1);
myErr = AddMovieResource(myDstMovie, myRefNum, NULL, NULL);
```

And at this point, we can safely close the destination movie file:

```
CloseMovieFile(myRefNum);
```

We also need to close down the instance of the standard image compression dialog component that's been doing all this work for us:

```
if (myComponent != NULL)
    CloseComponent(myComponent);
```

Finally, let's restore the source movie to its original graphics world and movie time, and dispose of the offscreen graphics world:

```
if (mySrcMovie != NULL) {
    // restore the source movie's original graphics port and device
    SetMovieGWorld(mySrcMovie, mySavedPort, mySavedDevice);
    // restore the source movie's original movie time
    SetMovieTimeValue(mySrcMovie, myOrigMovieTime);
}
// restore the original graphics port and device
SetGWorld(mySavedPort, mySavedDevice);
// delete the GWorld we were drawing frames into
if (myImageWorld != NULL)
    DisposeGWorld(myImageWorld);
```

We have now successfully compressed our image sequence. (Whew!) I won't bother to give the entire listing of the `QTCmpr_CompressSequence` function, as it would run for several pages. You can find the complete story in the source file `QTCompress.c`.

Mac OS X Porting & Development Showcase

Mac OS X Making the Move

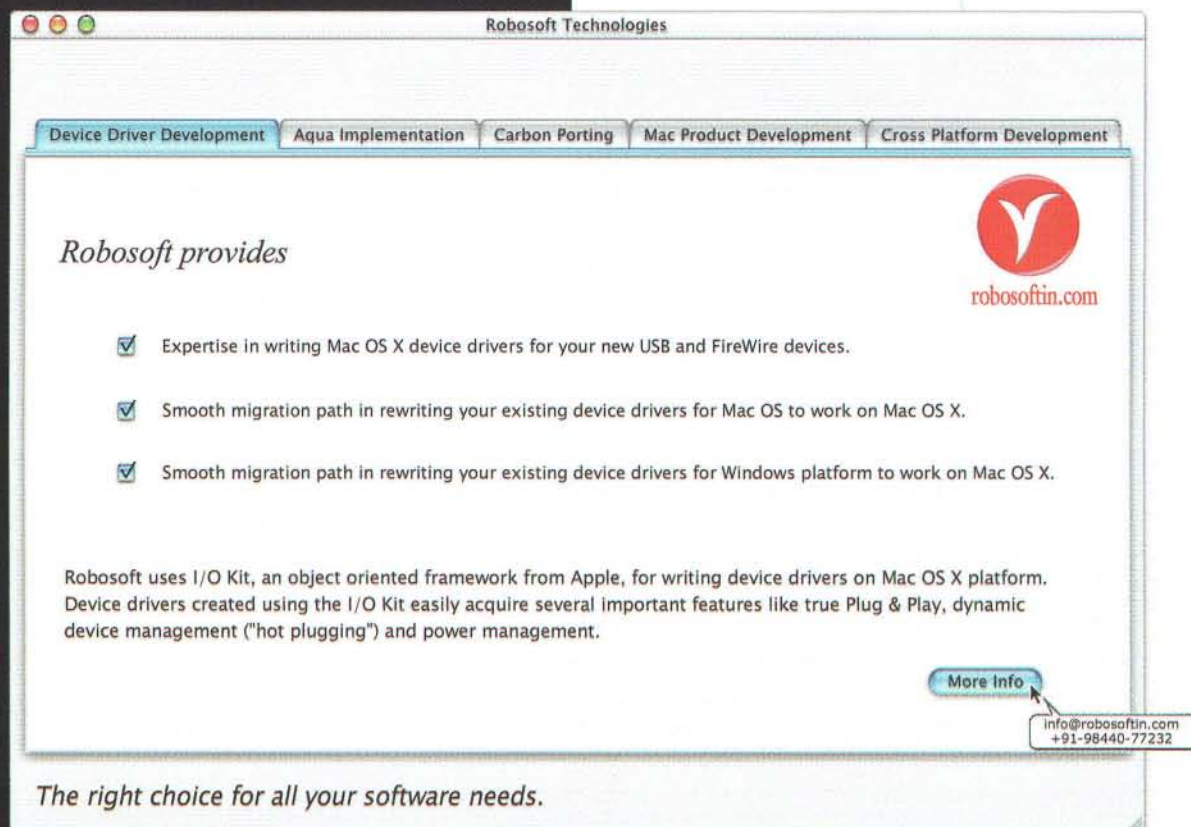
With **MAC OS X** just over the **HORIZON**,
Mac *developers* everywhere have a
major need for **CARBON** and **COCOA**
application porting, **AQUA** user interface
implementation, and **DRIVER** development
services. Toward that end, several
high-quality **SOFTWARE** engineering
firms, in association with the **APPLE**
DEVELOPER CONNECTION, are offering these
services at very *attractive* discounts
to all **ADC** Select and Premier members

The following pages
are those vendors that
are part of the Mac OS X
Porting & Development Showcase

Device Driver Development?

Special
Discounts
for
ADC Members

Get the most **experienced**
Mac product development company
working for **you.**



The screenshot shows a web browser window titled "Robosoft Technologies". The navigation bar includes links for "Device Driver Development", "Aqua Implementation", "Carbon Porting", "Mac Product Development", and "Cross Platform Development". The main content area features the Robosoft logo (a red circle with a white stylized 'Y') and the website URL "robosoftin.com". Below the logo, the text "Robosoft provides" is followed by a list of services with checkmarks:

- ✓ Expertise in writing Mac OS X device drivers for your new USB and FireWire devices.
- ✓ Smooth migration path in rewriting your existing device drivers for Mac OS to work on Mac OS X.
- ✓ Smooth migration path in rewriting your existing device drivers for Windows platform to work on Mac OS X.

Below the list, a paragraph states: "Robosoft uses I/O Kit, an object oriented framework from Apple, for writing device drivers on Mac OS X platform. Device drivers created using the I/O Kit easily acquire several important features like true Plug & Play, dynamic device management ("hot plugging") and power management."

A "More Info" button is located at the bottom right of the content area, with a tooltip displaying the contact information: "info@robosoftin.com" and "+91-98440-77232".

The footer of the screenshot reads: "The right choice for all your software needs."

Also contact us for:

- ✓ Porting your existing Windows products to the Macintosh Platform.
- ✓ Porting (Carbonizing) your existing Mac OS products to Mac OS X.
- ✓ Providing integrated eBusiness solutions using WebObjects.
- ✓ Designing and Developing Cross Platform products.

**Partners
in Product
Development**



get to

Mac OS X

fast!

*With the experienced
Mac OS development team.*

Red Rock Software specializes in Macintosh software development. Our team of senior engineers averages over 10 years of development experience on the Macintosh platform. Our experience and expertise has gained us the trust of companies like *Apple Computer, Iomega Corporation* and *Sorenson Media*.

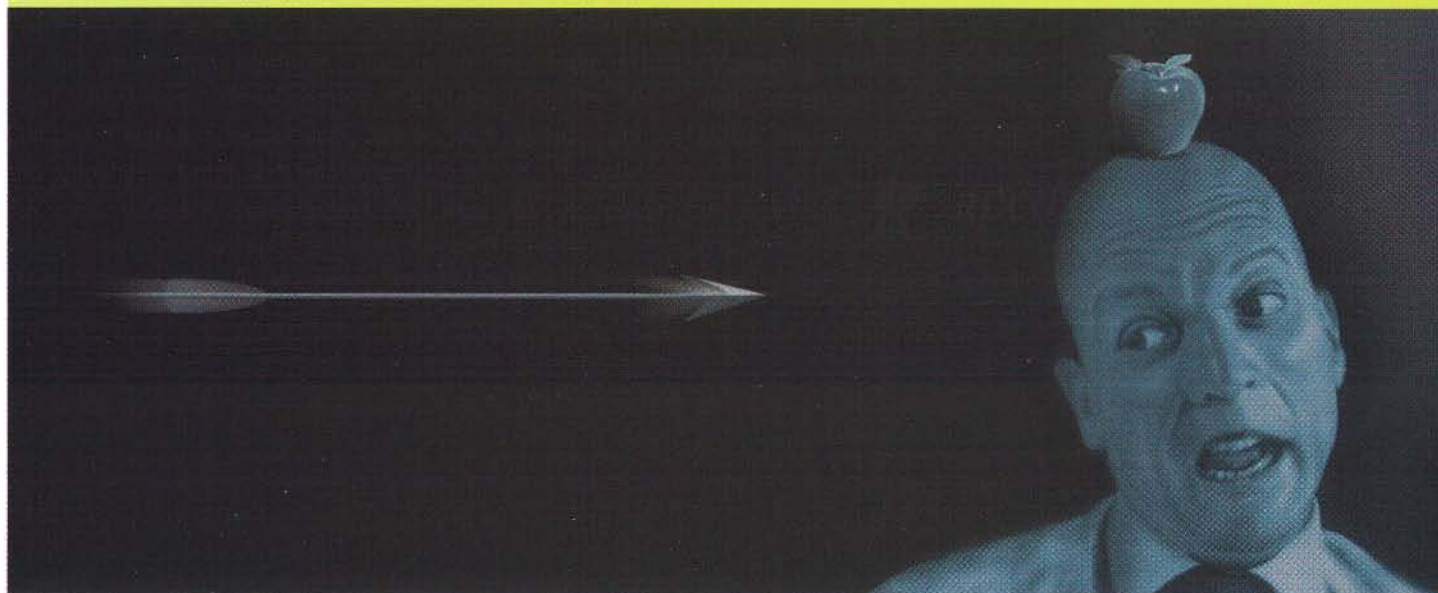
If you are looking to make your product compatible with Mac OS X quickly, let our experts help.

- Aqua Interface Implementation
- OS X Carbon Porting
- OS X Native Cocoa Application Development

R E D R O C K
s o f t w a r e

Call Red Rock at **888.689.3038**
or visit us online
at **www.redrocksw.com**

SOME THINGS ARE
BEST LEFT TO EXPERTS



Don't want to miss the mark while moving to **Mac OS X**. Come to RecoSoft. We're the experts in developing software for the Macintosh platform. Our quiver of skills include Macintosh Application Design & Development, Cross Platform (Mac/Win/Linux) Development Services, Double Byte and Localization Solutions, Software Quality Assurance & Contract Programming Services. We've also hit bulls eye in AQUA implementation, CARBON porting, QuickTime development and Japanese Language implementations. So, if your aim is to get the best software development for **Mac OS X**, target the experts.



RECOSOFT CORPORATION

RecoSoft Corporation, Osaka, Japan. Tel.: +81-6-6443-0015 Fax: +81-6-6443-1458 E-mail: info@recoSoft.com Website: www.recoSoft.com

- ☐ Eat your vegetables.
- ☐ Exercise every day.
- ☒ Port to Mac OS X.
- ☐ Call your mom.

All of these are good for you.
We can make one of them easy.

Since 1989, The Omni Group has worked with the technologies that have been refined into Mac OS X.

Moving to Mac OS X is a big step for your company, and you need consultants who can help you both plan how best to make the transition and follow whatever path you choose.

That's been our business for years. No matter what kind of product you have, we can get it up under OS X, fast:

- Real games: We ported id's Doom and Quake games to NEXTSTEP and OS X. Quake 2 took us a week.
- Big apps: We ported Adobe's FrameMaker to NEXTSTEP and Sun's Concurrency to OpenStep/Solaris.
- Big libraries: We ported the Oracle 8 client libraries which Apple ships today in OS X Server.
- Serious drivers: We ported 3dfx's Glide and wrote Voodoo2 and Rendition drivers for OS X Server.
We've written new mouse drivers for OS X Server and joystick drivers for OpenStep.
- New apps: We wrote OmniWeb, the only native OS X web browser, and OmniPDF, the native Acrobat viewer for OS X.

Mac OS X is what we do. Let us help you do it, too.

THE OMNI GROUP



2707 Northeast Blakeley Street
Seattle, Washington 98105-3118
www.omnigroup.com/consulting

sales@omnigroup.com
800.315.OMNI x201
206.523.4152 x201

Programming at the Speed of Light

Aqua

Cross-Platform

Device Drivers

TCP / IP

Graphics

Plug-Ins

Mac/Unix/Win32

Ports

Carbon / OS X



Software Development
Outsourcing Since 1990

415.491.2200
www.shadetreeinc.com

For our reference constant, we're passing the address of a local variable of type `OSErr`. The idea is that we can undertake other processing inside the `for` loop as long as the value of that variable is set to a known value; the completion routine, when it is triggered, will be responsible for changing that value, as we'll see shortly. Before we initiate the asynchronous compression operation, we'll set `myICMComplProcErr` to a default value:

```
myICMComplProcErr = kAsyncDefaultValue;
```

Performing Asynchronous Compression

As we've seen, we can begin an asynchronous compression operation by calling `SCCompressSequenceFrameAsync`, like this:

```
myErr = SCCompressSequenceFrameAsync(myComponent, myPixMap,
    &myRect, &myCompressedData, &myDataSize, &mySyncFlag,
    myICMComplProcPtr);
```

The parameters here are exactly those that we earlier passed to `SCCompressSequenceFrame`, with the additional final parameter `myICMComplProcPtr`. As with all asynchronous functions, `SCCompressSequenceFrameAsync` returns immediately. When the compressor is finished with the compression operation, our completion routine will be executed. Listing 8 shows our application's completion routine.

Listing 8: Signaling the end of a compression operation

```
static PASCAL_RTN void QTCmpr_CompletionProc
    (OSErr theResult, short theFlags, long theRefCon)
{
    OSErr *myErrPtr = NULL;
    if (theFlags & codecCompletionDest) {
        myErrPtr = (OSErr *)theRefCon;
        if (myErrPtr != NULL)
            *myErrPtr = theResult;
    }
}
```

When the `codecCompletionDest` flag is set in the `theFlags` parameter, we know that the compression operation has finished successfully. At that point, we can copy the result code passed to our completion routine into the location specified by `theRefCon`. This effectively changes the value of the `myICMComplProcErr` variable in our `for` loop, where we are executing this `while` loop until that value changes:

```
while (myICMComplProcErr == kAsyncDefaultValue) {
    EventRecord myEvent;
    WaitNextEvent(0, &myEvent, 60, NULL);
    SCAsyncIdle(myComponent);
}
```

As you can see, we're repeatedly calling `WaitNextEvent` to yield time to other processes and `SCAAsyncIdle` to yield time to the compressor component. I'll leave it as another exercise to add the code necessary to handle application events (for example, update events).

Weighing the Benefits

While adding support for asynchronous compression is not too terribly complicated, you might be wondering whether it's worth the trouble. In other words, are we going to see appreciable performance gains, or at least significant user

experience improvements? Sadly, the answer is "no", at least at the moment. First of all, not all compression components support asynchronous operation. (Indeed, at this point, there is only one Apple-supplied codec capable of performing asynchronous compression, the H.263 codec.) You can still invoke compression components using `SCCompressSequenceFrameAsync`, but `SCCompressSequenceFrameAsync` will not return to the caller until the compression is complete and the completion routine has been executed. In other words, it will run synchronously. Second, and perhaps more important, even codecs that are able to compress asynchronously will not show any real improvement when running on a single-processor machine.

CONCLUSION

The standard image compression dialog component is a prime example of the kind of QuickTime APIs that I like best: it provides a high-level toolbox for handling a wide variety of typical image compression tasks, allowing us to compress both individual images and sequences of images. Moreover, it gives us a standard user interface for eliciting compression settings from the user (rather in the same way that the Standard File Package or the Navigation Services provide a standard user interface for eliciting files from the user).

CREDITS

Much of the code in the function `QTCmpr_CompressSequence` is based on code in the earlier sample code package called "ConvertToMovie Jr." (which was itself based on an even earlier sample called "ConvertToMovie").

MT

BasePlant

200 classes to enhance PowerPlant™

- AppleEvents
- AppleScript
- Collection Manager
- Data Browser
- Disk Item Filter
- Folder Parser
- Icon Services
- Multiprocessing
- Preferences Manager
- Starting Point
- Window Proxy

And Much More!

\$149 For commercial, in-house development
\$35 For personal freeware and shareware

CAFEDERIC.COM

by Bob Boonstra, Westford, MA

TRILITE

Tic-Tac-Toe is a trivial game. There are less than 9! possible games, far fewer if symmetry is taken into account, and certainly few enough for the outcome to be calculated in advance. But there is a variant of Tic-Tac-Toe that allows many more possible move sequences, and for which there may or may not be a guaranteed winning solution. This month you are going to have an opportunity to compete in the game of Trilite against your Challenge peers.

Trilite is like Tic-Tac-Toe in the sense that it is played on a 3x3 board, where two players alternate occupying squares with the objective of occupying three positions in a row. It differs from Tic-Tac-Toe in that a player may occupy only three positions at a time. When a player occupies a fourth position, one of the three previously occupied positions, the one that has been occupied the longest, becomes vacant. So after any move, there are always three vacant positions on the board, and one more that is about to become unoccupied when the current player occupies one of the three vacant positions. Sounds simple, right?

The prototype for the code you should write is:

```
typedef enum { /* numbering system for Board positions */
    kNoPosition=-1,
    kTopLeft=0, kTopCenter, kTopRight,
    kCenterLeft, kCenter, kCenterRight,
    kBottomLeft, kBottomCenter, kBottomRight
} BoardPosition;

typedef enum { /* possible values for a Board Position */
    kEmpty=-1,
    kPlayer1Staying=0, kPlayer1Disappearing,
    kPlayer2Staying, kPlayer2Disappearing
} PositionValue;

typedef PositionValue Board[9]; /* state of the Board */

BoardPosition PlayTrilite(
    const Board triliteBoard, /* current state of the Board */
    BoardPosition opponentPreviousPlay,
    /* the BoardPosition your opponent last played */
    int playerNumber, /* 1 if you are player 1, 2 if you are player 2 */
    Boolean newGame /* true the first time you are called for a new game */
);
```

For each game of Trilite, your `PlayTrilite` routine and that of your opponent will be called alternately until one of you wins by occupying three positions in a row, horizontally, vertically, or diagonally. The first time `PlayTrilite` is called for a new game, `newGame` will be set to `TRUE`. When `newGame` is `TRUE`, `playerNumber` will indicate whether you are the first (`playerNumber==1`) or second (`playerNumber==2`) player. Each time `PlayTrilite` is called, the `BoardPosition` last occupied by your opponent will be provided as `opponentPreviousPlay`. Finally, the current state of the `Board` will be provided to you as `triliteBoard`.

Trilite board positions have five possible values. Unoccupied positions have the value `kEmpty`. Positions occupied by player 1 have the value `kPlayer1Staying` or `kPlayer1Disappearing`, with the latter value distinguishing positions that will become empty following player 1's next move. Similarly, positions occupied by player 2 have the value `kPlayer2Staying` or `kPlayer2Disappearing`.

A sequence of moves works like this. Suppose the game has been going on for at least three pairs of turns, and it is player 1's turn to play. The Board will have six occupied positions, three by player 1 and three by player 2. One position for each player will be marked as "disappearing" on the next move. Player 1 will occupy one of the three remaining unoccupied positions, and — at the same time — the `kPlayer1Disappearing` position will become `kEmpty`. If player 1 now occupies three positions in a row, s/he is the winner. Otherwise, player 2 then occupies one of the three empty positions and the `kPlayer2Disappearing` position becomes `kEmpty`. Note that a player may not reoccupy the position about to disappear — the opponent is the first player with a chance to occupy that position. The astute reader might detect one element of a potential game strategy here.

Entries will compete against one another in a tournament structured so that each entry plays each other entry an even number of times, half playing first, and half playing second. If the number of entries is large, some other fair tournament scheme will be used. A game will be considered drawn when a time limit and a move count limit, not specified as part of the problem statement, are exceeded.

The winner will be the entry that scores the most points, where each game won is worth 1000 points, each game drawn is worth 500 points, and 1 point is deducted for each millisecond of execution time. The Challenge prize will be divided between the overall winner and the best scoring entry from a contestant that has not won the Challenge recently.

Your code and data must live in an application heap of 40MB. Any nontrivial tables used by your solution must be calculated at run time. Any entry that precalculates any significant part of the solution will be disqualified.

Those of you interested in experimenting with Trilite might want to check out the shareware game by John Mauro, at <http://screch.cs.alfred.edu/~maurojc/software/software.html#Trilite>.

This will be a native PowerPC Challenge, using the CodeWarrior Pro 6 environment. Solutions may be coded in C, C++, or Pascal. You can also provide a solution in Java, provided you also provide a test driver equivalent to the C code provided on the web for this problem.

// Like most Mac developers,
// I easily spend 12 hours a day

// staring at line after line of C++ code in tiny, 9 point Monaco.
// Sometimes it makes my eyes feel like they're on fire.

{
So the **last thing** I need is some
fuzzy monitor that adds to my headaches.
}

/***** begin excitement *****/

// That's why I'm so **jazzed** about this SGI monitor.

// Its ultra-high resolution = 1600 x 1024 and dpi = 110,
// giving me razor-sharp contrast.
// And the high refresh rate is
// **perfect** for poring through lines of code.

// At first, I was amazed at the clarity, the fine details that emerged.

{
It was like seeing things for the first time.
}

// Later, though, I learned to appreciate the **wide aspect ratio** [= 16:10],
// with a generous 17.3 inches of viewing area. SGI's 1600SW
// lets me have **all my documents** viewable at once, and it's
// a **flat panel** so it fits on my desk with room to spare.

// From the moment I saw this thing I was hooked. You will be, too.

{
Especially when you find out
how affordable it is.
}

// **Check it out.**

/***** end excitement *****/



Michael Whittingham
V.P., Engineering
Wired, Inc.

www.sgi.com/flatpanel

This advertisement was written by an actual engineer. This is the first time in 7 months that he's been seen during the day.
© 2000 Silicon Graphics, Inc. All rights reserved. SGI 1600SW and SGI are registered trademarks of Silicon Graphics, Inc.
Mac is a registered trademark of Apple Computer, Inc. For more information, visit our website or call 1-888-SGI-7373.



sgi[™]

Scripter 2

with **ScriptBASE**

NEW!
Version 2.2

**Tap the power of AppleScript
with Main Event's Scripter!**

**For professionals and novices
Webmasters and solution providers**

**No matter what your experience,
Scripter makes it easier!**

BEGINNER AT SCRIPTING?

- Unique command-builders help you learn to assemble commands
- Built-in tools for experimenting
- Shortcuts to speed scripting

EXPERIENCED WITH APPLESCRIPT?

- Unmatched single-step interactive debugging
- Watch and modify global and local variables while stepping
- Debug messages sent to script applications
- Includes ScriptBase to integrate scripting scenarios

*Scripter received MacWeek's 5-diamond rating – Twice!
"Scripter's virtuoso display of AppleScripting savvy and
debugging wizardry make it the best environment we've
found for learning or creating AppleScript scripts."*

—MacWeek

**Make AppleScript and your
applications work for you!**



Main Event
PO Box 21470
Washington, DC 20009
Tel: 202-298-9595
Sales: 800-616-8320
info@mainevent.com
www.mainevent.com

THREE MONTHS AGO WINNER

Three people entered the November FreeCell Challenge, where contestants had to write code to solve the FreeCell solitaire puzzle. FreeCell requires players to move cards from eight tableaux to four home piles in ascending order based on suit, but it also provides four "free cells" where cards may be stored temporarily. Congratulations to **Ernst Munter (Kanata, Ontario)** for his victory in this Programmer's Challenge.

Ernst's entry performs a depth-first search of possible moves, enumerated by the `GenerateMoveList` routine. Moves are assigned a value that combines a heuristic weight assigned a priori to the type of move (e.g., `kFreeToHome`), a measure of the degree to which the cards in a tableau are in the correct order, and the presence in a tableau of cards that could be moved home. The code (`IsNotRedundant`) avoids moves that return a card to the position it occupied previously when no intervening move would have made the return nonredundant. A key to the speed of Ernst's entry is the way it avoids looping back into a previously encountered configuration. The `Execute` routine computes a hash value for the game state resulting from a prospective move and compares that hash value to that of previously encountered game states. If the prospective move results in a previously encountered state, the move is rejected. Assuming a move is not redundant, the move is made and a new set of possible moves is generated. The move search gives up and restarts if it is forced to backtrack too many times, using the list of previously encountered states to ensure that a different search path results.

As the top-placing contestant without a previous Challenge win, **Greg Sadetsky** wins a share of this month's Developer Depot merchandise credit prize. His second place solution also keeps track of past game states, but in a very large array instead of in a hash value. Greg employs a number of devices to reduce the storage required, but the resulting logic for detecting a repeat game state is more complex and time consuming. Greg's entry generates move sequences that are about 50% longer on average than those generated by Ernst's entry. It cuts off the search after 10 seconds, the point at which the time penalty exceeded the point value of solving the hand. As a result, his solution gave up on about 6% of the test cases.

The third entry I received this month was a recursive solution only slightly slower than the winning entry, but it crashed for 9 of the test cases. Even after I increased the heap and stack sizes significantly, the code crashed with heap corruption after apparently entering a recursion loop. To measure performance on the remaining cases, I needed to modify the test code to bypass the problematic hands and, for that reason, the entry was disqualified.

I tested the entries to this Challenge with more than 20,000 deals, including roughly one third of the 32,000 deals included in the Linux `xfreecell` package, 10,000 random deals, and a few manually constructed deals. Ernst's solution solved all but two of the test cases, both of which were a single deal that is known to be unsolvable. His solution required just over three minutes to

LA to New York

(and Seattle and Miami)

in 150 milliseconds or less*



Call for a TeraMedia® demo

TeraGlobal
COMMUNICATIONS

(858) 404.5500 ext 5660
www.teraglobal.com
sales@teraglobal.com



Interact. Collaborate. It's highest-quality multi-point video conferencing and true real-time multimedia collaboration. It's all done in software. Any number of users. Over any distance. At any time. In real time. It flies, you don't.

* Actual video latency varies with bandwidth. In most cases, TeraMedia® latency is below the threshold of human perception.

TeraMedia®
by TeraGlobal

run the entire set of tests, and generated an average of 156 moves to solve each deal. As you can see in the table below, a small number of test cases required more than 1500 moves to solve — the most complicated deal, excluding the ones that could not be solved, required 1863 moves.

	>100	>500	>1000	>1500	No
	Moves	Moves	Moves	Moves	Solution
	(# of cases)	(# of cases)	(# of cases)	(# of cases)	(# of cases)
Ernst Munter	18190	444	24	7	2
Greg Sadetsky	19410	784	49	11	1274
C. W.	20303	0	0	0	278

The table below lists, for each of the solutions submitted, the number of test cases solved by each entry, the total execution time, the number of points earned, and the number of moves generated to solve the entire test suite. It also provides the code size, data size, and programming language used for each entry. As usual, the number in parentheses after the entrant's name is the total number of Challenge points earned in all Challenges prior to this one. The solution marked with an asterisk was disqualified for reasons explained above.

Name	Test Cases	Time	Points	Moves	Code	Data	Lang
	Solved	Unsolved (secs)	x100000	x1000	Size	Size	
Ernst Munter (681)20694	2	181.1	206.8	3220	9800	1793	C++
Greg Sadetsky (2)19422	1274	24399.2	169.8	4705	8156	18.31M	C
C. W. (*)	20409	278	198.2	203.9	4103	7276	1858

TOP CONTESTANTS ...

Listed here are the Top Contestants for the Programmer's Challenge, including everyone who has accumulated 20 or more points during the past two years. The numbers below include points awarded over the 24 most recent contests, including points earned by this month's entrants.

Rank	Name	Points	Rank	Name	Points
1.	Munter, Ernst	271	5.	Boring, Randy	52
2.	Saxton, Tom	76	6.	Shearer, Rob	48
3.	Maurer, Sebastian	68	7.	Taylor, Jonathan	36
4.	Rieken, Willeke	65	8.	Willborg, Claes	29

... AND THE TOP CONTESTANTS LOOKING FOR A RECENT WIN

In order to give some recognition to other participants in the Challenge, we also list the high scores for contestants who have accumulated points without taking first place in a Challenge during the past two years. Listed here are all of those contestants who have accumulated 6 or more points during the past two years.

Rank	Name	Points	Rank	Name	Points
9.	Downs, Andrew	12	17.	Strout, Joe	10
10.	Jones, Dennis	12	18.	Hala, Ladislav	7
11.	Day, Mark	10	19.	Miller, Mike	7
12.	Duga, Brady	10	15.	Nicolle, Ludovic	7
13.	Fazekas, Miklos	10	21.	Schotsman, Jan	7
14.	Flowers, Sue	10	22.	Widyatama, Yudhi	7
15.	Sadetsky, Gregory	10	23.	Heithcock, JG	6
16.	Selengut, Jared	10			

There are three ways to earn points: (1) scoring in the top 5 of any Challenge, (2) being the first person to find a bug in a published winning solution or, (3) being the first person to suggest a Challenge that I use. The points you can win are:

1st place	20 points
2nd place	10 points
3rd place	7 points
4th place	4 points
5th place	2 points
finding bug	2 points
suggesting Challenge	2 points

Here is Ernst's winning FreeCell solution:

FreeCell.cp

Copyright © 2000

Ernst Munter, Kanata, ON, Canada

/*

Solves FreeCell games by a guided trial and error search.

At each stage, all possible moves are listed, ranked according to a fixed heuristic which prefers moves towards home, and towards aggregating strings of alternating colors on

DATA RECOVERY: 800-440-1904

"Their incredibly kind staff focus on getting the job done as well as their customer's feelings. All experiences should be so pleasant."

—Neil Ticktin, Publisher
MacTech Magazine

7 Good Reasons to Choose DriveSavers



1. Fastest, most successful data recovery service available.
2. Retrieve recovered data instantly with DATAEXPRESS™ over high speed secured Internet lines.
3. Authorized to perform Data Recovery on all Apple computers and hard drives.
4. 24-hour, onsite, and weekend services.
5. Advanced, proprietary recovery techniques.
6. Featured by CNN, BBC, Forbes. Also in Mac World, Mac Addict, Popular Mechanics, and many others.
7. Federal and State Contracts (GSA, CMAS).



"We Can Save It!"

INTL: 415-382-2000
www.drivesavers.com

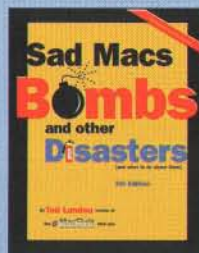
DriveSavers – Your Data Recovery Solution!

©2000 DRIVESAVERS, INC. 400 BEL MARIN KEYS BLVD., NOVATO, CA 94949, INTL: 415-382-2000, FAX: 415-883-0780

Peachpit Press

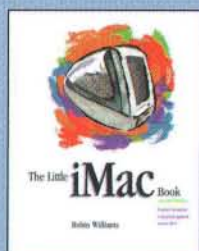
Get the most from your Mac

Great
Books!



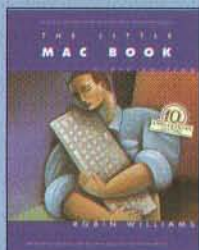
Sad Macs, Bombs, and Other Disasters, 4th Edition

By Ted Landau
ISBN 0-201-69963-X
980 pages • \$34.99 u.s.



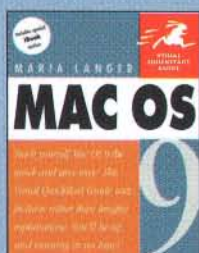
The Little iMac Book, Second Edition

By Robin Williams
ISBN 0-201-70446-3
312 pages • \$17.99 u.s.



The Little Mac Book, Sixth Edition

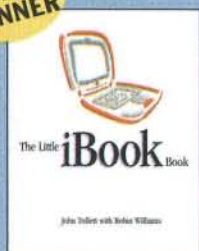
By Robin Williams
ISBN 0-201-35433-0
448 pages • \$19.99 u.s.



Mac OS 9: Visual QuickStart Guide

By Maria Langer
ISBN 0-201-70004-2
376 pages • \$17.99 u.s.

AWARD
WINNER



The Little iBook Book

By John Tollett and Robin Williams
ISBN 0-201-70093-X
232 pages • \$18.99 u.s.

The hottest graphics and Web books at great prices!

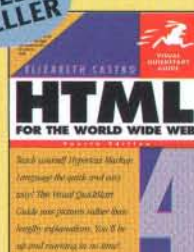
NEW



Flash 5 for Windows and Macintosh: Visual QuickStart Guide

By Katherine Ulrich
ISBN 0-201-71614-3
544 pages • \$21.99 u.s.

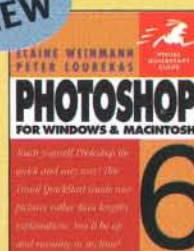
BEST
SELLER



HTML for the World Wide Web, Fourth Edition: Visual QuickStart Guide

By Elizabeth Castro
ISBN 0-201-35493-4
384 pages • \$19.99 u.s.

NEW



Photoshop 6 for Windows and Macintosh: Visual QuickStart Guide

By Elaine Weinmann and Peter Lourekas
ISBN 0-201-71309-8
448 pages • \$21.99 u.s.



Peachpit Press

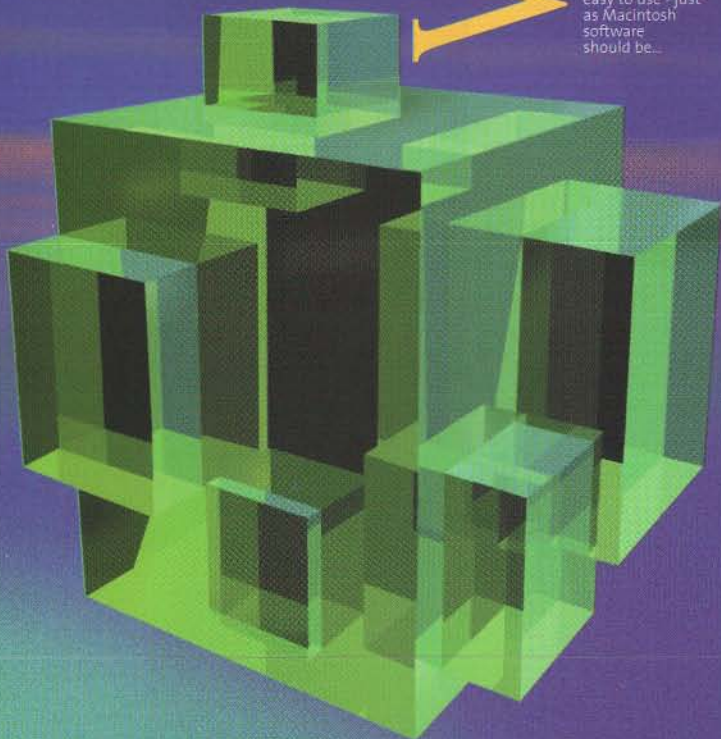
Peachpit Press books are available at quality bookstores everywhere. For a list of bookstores in your area see www.peachpit.com/order/stores.html

The Version Control Tool for CodeWarrior Developers

VOODOO Server



The successor of
our award-winning
VOODOO. More
powerful and still
easy to use - just
as Macintosh
software
should be...



www.unisoftwareplus.com
voodoo@unisoftwareplus.com

Available
@ **DEPOT**®
877-DEPOT-A-Now or 800-494-9797
Fax: 905-494-9798
<http://www.unidepot.com>
order@unidepot.com

the tableau.

All reached states are recorded in a database to avoid loops. The hash method to compress states takes care of some redundancies; for example it does not care which column a particular set of cards is in, and it distinguishes cards only bycolor, not suit.

If a search is making little progress, it is cut off after a specific number of undo steps, and a fresh search started. The same happens when the maximum number of moves has been reached. The new search still respects the accumulated database of previously seen states, and so is forced to take a different path, improving its chances.

The resulting move sequences are not optimal, and certainly not elegant. The search also does not include macro moves (moving columns of several cards). I tried this but it was counter-productive: by listing the macro moves, the move lists became longer, and more false paths ended up being explored.

Version 2 changes

- replaced the STL set<> with a simpler, faster custom set;
- replaced qsort (of move lists) with an integrated custom heap sort;
- policy constants tuned.

Version 3 change

Reduced the amount of redundant moves by scanning back through the move stack to avoid any move that would simple put a card back where it was earlier. Such moves are truly redundant if the to- and from- card positions were not used by intermediate moves of other cards. This strategy improved both time, and average number of moves to solve, by about 18%.

*/

```
#include "FreeCell.h"
#define NDEBUG
#include <assert.h>
#include <string.h> // for memset()

#define VERSION 3

// I need to have the suits in alternating red-black order.
enum {
    mySpade=0, myHeart=16, myClub=32, myDiamond=48, mySuits=48,
    myRed=16,

    myNull=0, myA=1, my2, my3, my4, my5, my6, my7, my8, my9, myT, myJ, myQ, myK,
    mySpots=15, kSignificant=myRed|mySpots
};

typedef unsigned char MyCard; // 2 bits suit + 4 bits spot
typedef unsigned char uchar;
typedef unsigned long ulong;
typedef unsigned short ushort;

enum {
    kFreeCell=0, // a single set of 16 card stacks defines the tableau
    kTableau=4, // card stack offsets
    kHome=12, // home must be last group

    kAvgMoveListLength=16, // just an estimate

    // Policy constants affect the order in which moves are tried:
    kFreeToHome=10000,
    kTableauToHome=10000,
    kTableauToTableau=2000,
    kFreeToTableau=500,
    kFreeToEmptyTableau=500,
    kTableauToEmptyTableau=50,
    kTableauToFree=24,
    kSrcPriority=2000,
    kBlockedOnly=0,

    kLongestPossibleMoveList=63, // actually no more than 31 have been
    observed
    kUndoLimitMul=16,
    kMaxRestartsDiv=65536
};

inline MyCard MakeCard(int spot, int suit)
{ return spot | (suit<<4); }
inline int MySuit(MyCard c) { return c>>4; }
```

struct CRC

// Standard CRC based hash method.


```
static struct CRC {
    enum {POLYNOMIAL=0x04c11db7L};
    ulong table[256];
    CRC()
    {
        long i,j,x;
        for (i=0;i<256;i++) {
            x=i<<24;
            for (j=0;j<8;j++) {
                if (x<0) x=(x<1) ^ POLYNOMIAL;
                else x=(x<1);
            }
            table[i]=x;
        }
    }
    ulong HashFunction(const uchar* ufrg,int frgLen,int type)
    const
    {
        // Uses CRC on length type, and all chars of a fragment
        ulong accum=table[frgLen];
        for (int i=0;i<frgLen;i++)
            accum=(accum<<8) ^ table[(accum>>24) ^
                (kSignificant & ufrg[i])];
        accum=(accum<<8) ^ table[(accum>>24) ^ type];
        return type + accum;
    }
} crc;
```

struct Legal

```
// A pair of lookup tables to indicate legality of placing one card upon another.
static struct Legal {
    bool redBlack[64][64]; // legal to put second card on (first) in
    tableau
    bool inSequence[64][64]; // legal to send second card home (first)
    Legal()
    // setup red-black and inSequence card lookup tables
    {
        for (int first=(myNull|mySpade);
            first<=(myK|myDiamond);first++)
        {
            for (int second=(myA|mySpade);
                second<=(myK|myDiamond);second++)
            {
                if ( ((mySpots & (first - second))==1) &&
                    ((myRed & (first ^ second))==myRed) )
                    redBlack[first][second]=true;
                // else =0;
                if ( ((mySpots & (second - first))==1) &&
                    ((mySuits & (first ^ second))==0) )
                    inSequence[first][second]=true;
                // else =0;
            }
        }
    }
} gLegal;
```

```
inline MyCard Convert2MyCard(const Card c)
// converts a "Card" defined in "FreeCell.h" to an instance of "MyCard"
{
    switch (c.suit)
    {
        case kSpade: return mySpade | c.spot;
        case kHeart: return myHeart | c.spot;
        case kDiamond: return myDiamond | c.spot;
        case kClub: return myClub | c.spot;
    }
    return 0;
}
```

struct CardStack

```
struct CardStack {
    // Generic card stack, serving for tableau, freecell, and home columns
    MyCard* SP;
    uchar stackType;
    MyCard cards[27]; // only 19 needed, struct is padded out to 32 bytes
    void Init(const Tableau* theTableau,int num,int type)
    {
        stackType=type;
        SP=cards;
        if (theTableau)
            for (int i=0;i<num;i++)
                *SP++=Convert2MyCard(theTableau->theCard[i]);
    }
    void InitHome(int suit)
    {

```

```
        stackType=kHome;
        SP=cards+1;
        cards[0]=MakeCard(myNull,suit); // null card of correct suit to build
    upon
    {
        MyCard TopCard() const {return SP[-1];}
        ulong Hash() const
        {
            return crc.HashFunction(cards,NumCards(),stackType);
        }
        bool IsEmpty() const {return SP==cards;}
        void Add(MyCard c)
        {
            assert(NumCards()<19);
            *SP++=c;
        }
        MyCard Remove()
        {
            assert(SP>cards);
            return *--SP;
        }
        int AllInOrder()
        // If the entire tableau stack is in order, returns numCards.
        // If not, this function returns 0.
        {
            int num=0;
            if (SP>cards)
            {
                num++;
                MyCard* c1=SP-1;
                while (c1>cards)
                {
                    MyCard* c2=c1-1;
                    if (!gLegal.redBlack[*c2][*c1])
                        return 0;
                    num++;
                    c1=c2;
                }
            }
            return num;
        }
    }
}
```

USBStuff

1-88-USB USB US
-or- (1-888-728-7287)

WorldWide Distributors of USB
and FireWire Parts,
Peripherals and Accessories

Hey Developers:

<http://www.usbstuff.com/developers.html>

1-877-4 HOTWIRE
-or- 1-877-446-8947

FireWireStuff


```

int NumInOrder()
// Returns the number of cards at the top of the stack which are in order.
{
    int num=0;
    if (SP>cards)
    {
        num++;
        MyCard* c1=SP-1;
        while (c1>cards)
        {
            MyCard* c2=c1-1;
            if (!gLegal.redBlack[*c2][*c1])
                break;
            num++;
            c1=c2;
        }
    }
    return num;
}

int SourcePriority(MyCard home[])
// Scans the stack including (or excluding) the top card, to set a priority value
// for the stack if it contains cards that could go home right away.
// kBlockedOnly=1 limits priority to blocked cards.
// Returns the priority value
{
    int srcPriority=0;
    MyCard* cp=cards;
    for (;cp<SP-kBlockedOnly;cp++)
    {
        MyCard c=*cp;
        for (int k=0;k<4;k++)
        {
            if (c==home[k])
                srcPriority+=kSrcPriority;
        }
    }
    return srcPriority;
}

int NumCards() const {return SP-cards;}
};

```

```

struct MyMove {
// My move is represented in a 32-bit ulong
    ulong gameValue:16; // value of this move or cardToMove
    ulong toPile:8;
    ulong fromPile:8;
    void Init(int from,int to,int val)
    {
        gameValue=val;
        toPile=to;
        fromPile=from;
    }
    void Clear() {fromPile=toPile=gameValue=0;}
    ulong IsValid() const {return Int();} // Null-move indicated by all-0
fields
    ulong FromPile() const {return fromPile;}
    ulong ToPile() const {return toPile;}
    void SetValue(MyCard c) {gameValue=c;}
    bool IsInverseOf(MyMove m) const {
        return ((fromPile == m.toPile) && (toPile ==
m.fromPile));
    }
    bool ToHome() const {return (toPile>=kHome);}
    void MoveCard(CardStack* stacks)
    {
        assert(stacks[fromPile].NumCards());
        assert(stacks[toPile].NumCards()<19);
        MyCard c=stacks[fromPile].Remove();
        stacks[toPile].Add(c);
    }
    void UndoMove(CardStack* stacks)
    {
        assert(stacks[toPile].NumCards());
        assert(stacks[fromPile].NumCards()<19);
        MyCard c=stacks[toPile].Remove();
        stacks[fromPile].Add(c);
    }
    void Convert(Move* m)
// Converts this instance of "MyMove" to a "Move" as defined in "FreeCell.h"
    {
        m->theSource = Source(fromPile-kFreeCell+dFreeCellA);

```

```

        m->theDestination = (toPile>=kHome) ? dHome:
        Destination(toPile-kFreeCell+dFreeCellA);
    }
    int Int() const {return *((int*)this);} // cast all three fields as
single int
};
typedef MyMove* MyMovePtr;

inline bool operator > (const MyMove & a,const MyMove & b)
{return a.Int() > b.Int();}

```

```

struct MoveHeap
//The custom heap for sorting moves.
struct MoveHeap {
    int heapSize;
    MyMove heapBase[kLongestPossibleMoveList];
    MoveHeap() : heapSize(0) {}
    int Size() const {return heapSize;}

    void Insert(MyMove k)
    {
        int i=++heapSize;
        int j=i>>1;
        MyMove z;
        while (j && ((z=heapBase[j]) > k))
        {
            heapBase[i]=z;
            i=j;
            j=i>>1;
        }
        heapBase[i]=k;
    }

    MyMove Pop()
    {
        MyMove rc=heapBase[1];
        MyMove k=heapBase[heapSize-];
        if (heapSize<=1) {
            heapBase[1]=k;
            return rc;
        }
        int i=1,j=2;
        while (j<=heapSize)
        {
            if ((j<heapSize)
&& (heapBase[j] > heapBase[j+1]))
                j++;
            if (heapBase[j] > k)
                break;
            heapBase[i]=heapBase[j];
            i=j;j+=j;
        }
        heapBase[i]=k;
        return rc;
    }
};

```

```

struct Bucket
//The set (MySet below) is implemented as a hash table of buckets.
// Each bucket can hold kBucketSize values, and can be extended indefinitely
// by linking to additional buckets.
enum {kBucketSize=17,kNumBuckets=1024};
struct Bucket {
    int numEntries;
    Bucket* link;
    ulong entry[kBucketSize];
// bucket size of 9 or 17 makes full use of allocated memory (CW 6)
    Bucket(ulong firstEntry) :
        numEntries(1),link(0) {entry[0]=firstEntry;}
    ~Bucket() {if (link) delete link;}
    void Insert(ulong x)
// Insert x only if x is not in the set already
    {
        Bucket* b=Find(x);
        if (b==0) return;
        b->Add(x);
    }
    Bucket* Find(ulong x)
// Scans this and linked buckets looking for x
// Returns 0 if found, returns this if not found
    {
        ulong* ep=entry+numEntries;
        do {

```


Installer VISE 7.1

for Mac OS X



www.mindvision.com


```

        if (*-ep == x) return 0;
    } while (ep>entry);
    if (link) return link->Find(x);
    return this;
}

void Add(ulong x)
{
    if (numEntries < kBucketSize)
        entry[numEntries++]=x;
    else
        link=new Bucket(x);
}

struct MySet {
// A set to record all states (represented by their hash value) which have occurred.
    Bucket* buckets[kNumBuckets];
    MySet() {memset(buckets,0,sizeof(buckets));}
    ~MySet() {
        for (int i=0;i<kNumBuckets;i++)
        {
            Bucket* b=buckets[i];
            if (b) delete b;
        }
    }
    void Insert(ulong x)
    {
        Bucket* b=buckets[x % kNumBuckets];
        if (b==0)
        {
            b=new Bucket(x);
            buckets[x % kNumBuckets]=b;
        } else b->Insert(x);
    }
    bool Find(ulong x)
    {
        Bucket* b=buckets[x % kNumBuckets];
        return (b && (0==b->Find(x)));
    }
};

```

struct MySet

```

struct MyGame {
// MyGame is the top level struct which holds all local data
    CardStack stacks[16]; // my version of the tableau, the current state
    ulong hashedState; // current state, compressed
    long numCardsOutstanding;
    MyMove* movePool; // single pool allocated for movelists
    MyMove* endMovePool;
    MyMovePtr* moveStack; // move stack tracks the history of executed
moves
    MyMovePtr* moveStackPointer;
    MyMovePtr* lastMoveStack;
    MyCard nextHome[4]; // next cards (1 per suit) to go home
    MySet stateSet; // all visited states are recorded in this set, as
hash values
    MyGame(long maxMoves) :
        movePool(new MyMove[kLongestPossibleMoveList+
            maxMoves*kAvgMoveListLength]),
        endMovePool(movePool+kLongestPossibleMoveList+
            maxMoves*kAvgMoveListLength),
        moveStack(new MyMovePtr[maxMoves]),
        moveStackPointer(moveStack),
        lastMoveStack(moveStack+maxMoves-1)
    {}

    ~MyGame() {
        delete [] moveStack;
        delete [] movePool;
    }

    void InitTableau(const Tableau theTableau[8])
    // Copies the initial tableau to the local representation
    {
        for (int tid=0;tid<8;tid++)
            stacks[tid+kTableau].Init(&theTableau[tid],
                7-tid/4,kTableau);
        numCardsOutstanding=52;
        for (int i=0;i<4;i++)
        {
            stacks[i+kFreeCell].Init(0,0,kFreeCell);
            stacks[i+kHome].InitHome(i);
            nextHome[i]=MakeCard(myA,i);
        }
        hashedState=Hash();
    }

    ulong Hash() const
    // Hashes the game state into a single 32-bit integer
    {
        const CardStack* cs=stacks;
        ulong h=cs->Hash();
        for (int i=1;i<16;i++,cs++)
            h ^= cs->Hash();
        return h;
    }
};

```

MyGame::Hash

MyGame::GenerateMoveList

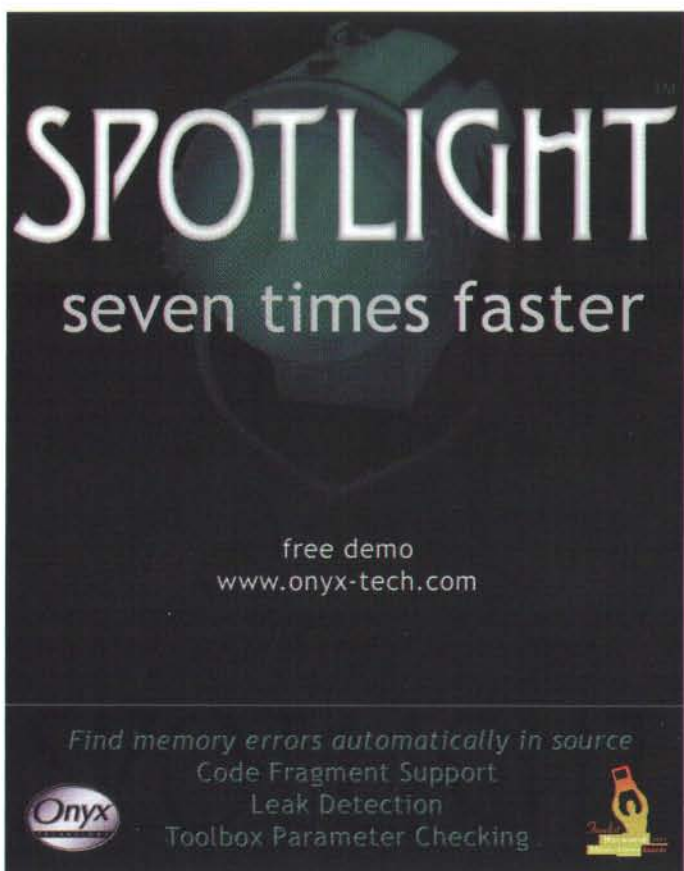
```

MyMove* GenerateMoveList(MyMove* mp)
{
// Lists all legal moves in a list, starting with a null-move;
// sorts the moves and returns the highest value move on the list
// Each move is given a "value" reflecting its relative merit.
    if (mp+kLongestPossibleMoveList >= endMovePool)
        return 0; // no room for movelist, should not really happen
// but if it does, we just have to backtrack

    MyMove m;
    MoveHeap heap;
    int src,dest;
    CardStack* srcPtr;
    CardStack* destPtr;
    int cardToMove,topCardDest,value,srcPriority;

    for (src=kFreeCell,srcPtr=stacks+src;
        src<kFreeCell+4;src++,srcPtr++)
    // from any freecell to: home, or tableau
    {
        if (srcPtr->IsEmpty()) continue;
        cardToMove=srcPtr->cards[0];
        srcPriority=srcPtr->SourcePriority(nextHome);
    }
}

```



SPOTLIGHT
seven times faster

free demo
www.onyx-tech.com

Find memory errors automatically in source
Code Fragment Support
Leak Detection
Toolbox Parameter Checking

Onyx
TECH


```

topCardDest=stacks[kHome+MySuit(cardToMove)].TopCard();
if (gLegal.inSequence[topCardDest][cardToMove])
    // to correct home
{
    value = kFreeToHome +
        srcPriority;
    m.Init(src,MySuit(topCardDest)+kHome,value);
    heap.Insert(m);
}

bool toEmptyFlag=true;
for (dest=kTableau,destPtr=stacks+dest;
    dest<kTableau+8;dest++,destPtr++)
// to every matching tableau
{
    if (destPtr->IsEmpty())
    {
        if (toEmptyFlag)
        {
            value = kFreeToEmptyTableau +
                (2<<(cardToMove&mySpots)) +
                srcPriority;
            m.Init(src,dest,value);
            heap.Insert(m);
            toEmptyFlag=false;
        }
        continue;
    }
    topCardDest=destPtr->TopCard();
    if (gLegal.redBlack[topCardDest][cardToMove])
    {
        value = kFreeToTableau +
            destPtr->AllInOrder() +
            srcPriority;
        m.Init(src,dest,value);
        heap.Insert(m);
    }
}

for (src=kTableau,srcPtr=stacks+src;
    src<kTableau+8;src++,srcPtr++)
// from any tableau to: freecell, home or tableau
{
    if (srcPtr->IsEmpty()) continue;
    int srcInOrder=srcPtr->AllInOrder();
    int longestInOrder=srcPtr->NumInOrder();
    srcPriority=srcPtr->SourcePriority(nextHome);
    int maxBlock=0;

    cardToMove=srcPtr->TopCard();// single card moves
    topCardDest=stacks[kHome+MySuit(cardToMove)].TopCard();
    if (gLegal.inSequence[topCardDest][cardToMove])
        // to matching home
    {
        value = kTableauToHome +
            srcPriority;
        m.Init(src,MySuit(topCardDest)+kHome,value);
        heap.Insert(m);
    }

    for (dest=kFreeCell,destPtr=stacks+dest;
        dest<kFreeCell+4;dest++,destPtr++)
// to first available freecell
    {
        if (destPtr->IsEmpty())
        {
            value = kTableauToFree -
                srcInOrder -
                4*longestInOrder +
                srcPriority;
            m.Init(src,dest,value);
            heap.Insert(m);
            break;
        }
    }
}

```

```

bool toEmptyFlag=true;
for (dest=kTableau,destPtr=stacks+dest;
    dest<kTableau+8;dest++,destPtr++)
// to every matching tableau
{
    if (src==dest) continue;
    if (destPtr->IsEmpty()) // to empty tableau
    {
        if (toEmptyFlag)
        {
            value = kTableauToEmptyTableau +
                srcInOrder +
                (2<<(mySuits & cardToMove)) +
                srcPriority;
            m.Init(src,dest,value);
            heap.Insert(m);
            toEmptyFlag=false;
        }
        continue;
    }

    topCardDest=destPtr->TopCard();
    if (gLegal.redBlack[topCardDest][cardToMove])
    {
        value = kTableauToTableau +
            destPtr->AllInOrder() -
            4*srcInOrder +
            srcPriority;
        m.Init(src,dest,value);
        heap.Insert(m);
    }
}

mp->Clear(); // puts a sentinel 0-move at the start of the
movelist

while (heap.Size()) // sorts moves from heap into the movelist space
    *++mp = heap.Pop();

return mp;
}

```

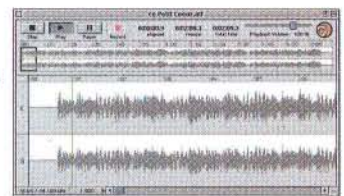
Felt Tip Sound Studio



Record and Edit Audio
with a sound editor designed for the Mac.
Sound Studio will allow you to make quick edits with an interface as easy as a text editor. Add polish to recordings with fades, normalization, and edits. Create your own mixes. Transform them with effects.

Sound Studio features

- up to 16 bits, 2 channels, and 65 kHz
- up to 2 GB of audio
- AIFF, Sound Designer 2, WAVE, System 7 Sound, and QuickTime import
- edit with sample accuracy
- fade, amplify, normalize, and invert
- delay, echo, reverse, and swap channels
- smooth and emphasize
- resample and pitch shift
- snap to zero crossings, snap to grid



\$35
Sound Studio
download free 14-day trial
or order online at
www.felttip.com

QuickTime is a trademark, used under license.
price in US dollars
Felt Tip Software, 807 Keely Place, Philadelphia PA 19128-2326, USA


```

void PushMove(MyMove* m){
    *moveStackPointer++=m;
}

MyMove* PopMove()
{
    assert(moveStackPointer>moveStack);
    return *--moveStackPointer;
}

MyGame::Execute
{
    int Execute(MyMove* mp)
    {
        // Attempts to execute one move.
        // Return codes:
        // -2: failed, cannot push the last move because the move stack is full
        // -1: failed, would have reached a previous state
        // 0: success, final move and game solved
        // >0: normal execution succeeded
        MyMove m=*mp;
        stateSet.Insert(hashedList); // save last state in hashed state set

        if (moveStackPointer >= lastMoveStack)
            return -2;

        if (m.ToHome() && (numCardsOutstanding==1)) //The game is
solved.
        {
            PushMove(mp);
            return 0;
        }

        // do the move and compute a new hashed state
        ulong newHash=hashedState ^
            stacks[m.FromPile()].Hash() ^
            stacks[m.ToPile()].Hash();

        MyCard cardToMove=stacks[m.FromPile()].TopCard();
        m.MoveCard(stacks);

        newHash ^=
            stacks[m.FromPile()].Hash() ^
            stacks[m.ToPile()].Hash();

        if (stateSet.Find(newHash))
        {
            m.UndoMove(stacks);
            return -1;
        } else
        {
            hashedState=newHash; // record new hash value
            mp->SetValue(cardToMove);
            PushMove(mp);
            if (m.ToHome())
            {
                nextHome[m.ToPile()-kHome]++;
                numCardsOutstanding--;
            }
        }
        return 1;
    }

    MyMove* Undo()
    // Undoes the last stacked move, returns this move, or 0 if no move found
    {
        MyMove* mp=PopMove();
        if (mp==0) return mp;
        MyMove m=*mp;
        ulong newHash=hashedState ^
            stacks[m.FromPile()].Hash() ^
            stacks[m.ToPile()].Hash();

        m.UndoMove(stacks);
        if (m.ToHome())
        {
            nextHome[m.ToPile()-kHome]--;
            numCardsOutstanding++;
        }

        hashedState=newHash ^
            stacks[m.FromPile()].Hash() ^

```

```

            stacks[m.ToPile()].Hash();

        return mp;
    }

    long CopyMovesBack(Move theMoves[])
    // Scans movestack, converts MyMoves to Moves, and returns the number of moves
    {
        int numMoves=0;
        MyMovePtr* endMoveStack=moveStackPointer;
        for (MyMovePtr*
index=moveStack+1; index<endMoveStack; index++)
        {
            MyMove* mp=*index;
            mp->Convert(theMoves+numMoves);
            numMoves++;
        }
        return numMoves;
    }

    int IsNotRedundant(MyMove m)
    {
        int from=m.FromPile();
        int to=m.ToPile();
        MyCard cardToMove=stacks[from].TopCard();
        MyMovePtr* mps=moveStackPointer;
        while (mps>moveStack)
        {
            MyMove* oldMove=*mps;
            int oldFrom=oldMove->FromPile();
            int oldTo=oldMove->ToPile();
            MyCard oldCard=oldMove->gameValue;
            if (oldCard==cardToMove)
            {
                return ((oldTo^from) | (oldFrom^to));
            } else
            {
                if ((oldFrom==to) || (oldTo==to) || (oldFrom==from))
                    break;
            }
        }
        return 1;
    }

    long Solve(const Tableau theTableau[8], Move theMoves[], long
maxMoves)
    {
        // Solves the game by systematic depth-first exploration of the move tree
        // Several fresh starts are possible if the move stack is exhausted
        // or if the search seems to be stuck with a large number of backtracks
        // In any case, all visited states are recorded in the hashed state set,
        // and never entered twice. The hash is not perfect, and some states might
        // be accidentally excluded. It is hoped that there is always enough redundancy in
        // the possible solution sequences to allow an alternative solution to be found.
        int cycle=kMaxRestartsDiv/maxMoves, rc;
        do {
            int undoLimit=kUndoLimitMul*maxMoves;
            InitTableau(theTableau);
            moveStackPointer=moveStack;
            // Put a sentinel null move at start of move stack
            PushMove(0);
            MyMove* moveList=movePool;
            #if VERSION<3
            MyMove previousMove;
            previousMove.Clear();
            #endif
            get_new_movelist:
            MyMove* nextMove=GenerateMoveList(moveList);
            // moveList to nextMove defines a movelist which always starts with a 0-move
            // and is processed in order nextMove, nextMove-1, ... until 0-move is found
            for (;;)
            {
                while (nextMove && nextMove->IsValid())
                {
                    #if VERSION>=3
                    if (!IsNotRedundant(*nextMove))
                    #else
                    if (nextMove->IsInverseOf(previousMove))
                    #endif

```


**"Without a doubt, the Premiere Resource Editor
for the Mac OS ... A wealth of time-saving tools."**

– MacUser Magazine Eddy Awards

"A distinct improvement over Apple's ResEdit."

– MacTech Magazine

"Every Mac OS developer should own a copy of Resorcerer."

– Leonard Rosenthal, Aladdin Systems

**"Without Resorcerer, our localization efforts would look like a
Tower of Babel. Don't do product without it!"**

– Greg Galanos, CEO and President, Metrowerks

"Resorcerer's data template system is amazing."

– Bill Goodman, author of *Smaller Installer* and *Compact Pro*

"Resorcerer Rocks! Buy it, you will NOT regret it."

– Joe Zobkiw, author of *A Fragment of Your Imagination*

"Resorcerer will pay for itself many times over in saved time and effort."

– MacUser review

"The template that disassembles 'PICT's is awesome!"

– Bill Steinberg, author of *Pyro!* and *PBTools*

"Resorcerer proved indispensable in its own creation!"

– Doug McKenna, author of *Resorcerer*



Resorcerer® 2

Version 2.0

The Resource Editor for the Mac™ OS Wizard

ORDERING INFO

Requires System 7.0 or greater,
1.5MB RAM, CD-ROM

Standard price: \$256 (decimal)

Website price: \$128 - \$256

(Educational, quantity, or
other discounts available)

Includes: Electronic documentation
60-day Money-Back Guarantee
Domestic standard shipping

Payment: Check, PO's, or Visa/MC
Taxes: Colorado customers only

Extras (call, fax, or email us):
COD, FedEx, UPS Blue/Red,
International Shipping

MATHEMAESTHETICS, INC.
PO Box 298
Boulder, CO 80306-0298 USA
Phone: (303) 440-0707
Fax: (303) 440-0504
resorcerer@mathemaesthetics.com

New
in
2.0:

- Very fast, HFS browser for viewing file tree of all volumes
- Extensibility for new Resorcerer Apprentices (CFM plug-ins)
- New AppleScript Dictionary ('aete') Apprentice Editor
- MacOS 8 Appearance Manager-savvy Control Editor
- PowerPlant text traits and menu command support
- Complete AIFF sound file disassembly template
- Big-, little-, and even mixed-endian template parsing
- Auto-backup during file saves; folder attribute editing
- Ships with PowerPC native, fat, and 68K versions

- Fully supported; it's easier, faster, and more productive than ResEdit
- Safer memory-based, not disk-file-based, design and operation
- All file information and common commands in one easy-to-use window
- Compares resource files, and even **edits your data forks** as well
- Visible, accumulating, editable scrap
- Searches and opens/marks/selects resources by text content
- Makes global resource ID or type changes easily and safely
- Builds resource files from simple Rez-like scripts
- Most editors DeRez directly to the clipboard
- All graphic editors support screen-copying or partial screen-copying
- Hot-linking Value Converter for editing 32 bits in a dozen formats
- Its own 32-bit List Mgr can open and edit very large data structures
- Templates can pre- and post-process any arbitrary data structure
- Includes nearly 200 templates for common system resources
- TMPLs for Installer, MacApp, QT, Balloons, AppleEvent, GX, etc.
- Full integrated support for editing color dialogs and menus
- Try out balloons, 'icb's, lists and popups, even create C source code
- Integrated single-window Hex/Code Editor, with patching, searching
- Editors for cursors, versions, pictures, bundles, and lots more
- Relied on by thousands of Macintosh developers around the world

To order by credit card, or to get the latest news, bug fixes, updates, and apprentices, visit our website...

www.mathemaesthetics.com


```

{
    nextMove--;
    continue; // while
}
rc=Execute(nextMove);
if (rc==-1) // would have reached a previous state
{
    nextMove--; // use next best move in list

    undoLimit--;
    if (undoLimit<=0) // enough! let's restart
        goto restart_search;
    } else

    if (rc>0) // move was executed, get next movelist
    {
        moveList=1+nextMove;
        previousMove=*nextMove;
        goto get_new_movelist;
    } else

    if (rc==0) // copy moves back for the caller and return
        return CopyMovesBack(theMoves);

    else // else rc<=-2: move stack is full
    {
        goto restart_search;
    }

} // end while

// no move is possible, try to backtrack
do {
    MyMove* prevMove=Undo();
    if (!prevMove) // no solution!, stack is completely unwound
        return 0;
}

```

```

// try to use the last move:
nextMove = prevMove-1;
assert(nextMove>=movePool);
assert(nextMove<moveList);
} while (!nextMove->IsValid());

moveList=nextMove;
while ((moveList>=movePool) && (moveList-
>IsValid()))
    moveList--;
assert(moveList>=movePool);
}
restart_search::
    while (~cycle > 0); // restart only so many times
return 0; // then give up and return 0
};

```

```

long FreeCell( // returns the number of moves in theMoves[]
const Tableau theTableau[8],
Move theMoves[],
long maxMoves
) {
    MyGame* G=new MyGame(maxMoves);
    long numMoves=G->Solve(theTableau,theMoves,maxMoves);

    delete G;
    return numMoves;
}

```

FreeCell

17

Intelligent Web site Monitoring and Analysis Software

FUNNELWEB
ENTERPRISE

now for the Mac OSX

Any Server, Any platform, Any time

www.funnelweb4.com info@activeconcepts.com Toll free 1800 551 5282

Speed, intuitive user interface, accuracy and in-depth analysis have always made Funnel Web the intelligent choice for Web site analysis.

Now includes pdf output, incremental analysis, cluster analysis and streaming media reports.

Visit our booth #3240 in the MacTech Pavillion and save 40% when you purchase Funnel Web Enterprise. Show Special for one week only.

MacAddict's

YOU KNOW WHO YOU ARE

Call 1-888-771-6222
TO GET YOUR FREE ISSUE AND
FREE CD-ROM NOW!

**TO RECEIVE YOUR
FREE ISSUE AND
FREE CD-ROM OF**

MacAddict

AT NO RISK!

If you're pleased with what you see, we'll send you 11 more issues and 11 more CD-ROMs (12 in all) for only \$29.95. That's a 68% savings off the newsstand rate.

Offer good in U.S. only. Cover price is \$7.99. Canada: U.S. \$45, includes GST. Foreign: U.S. \$58. Prepaid in U.S. funds only. Please allow 6-8 weeks for delivery of first issue.

APPLE REVS: DV POWERBOOKS! GRAPHITE iBOOKS! AND FASTER G4S!

A better machine. A better magazine.

MacAddict

MAY/00

The **ultimate mac product slugfest!**

31 PRODUCTS BATTLE IT OUT!

Quark vs. InDesign
Illustrator vs. Freehand
GoLive vs. Dreamweaver
Palm vs. Handspring
Quake vs. Unreal
DSL vs. Cable
AND MANY MORE!

9 USES FOR YOUR OLD

software slugfest!

You Pick the Winner!

INCLUDING:

- Unreal Tournament and Quake III
- GoLive and Dreamweaver
- Audion and SoundJam MP and QDesign MVP
- Illustrator and Freehand
- Endnote and Outlook Express
- And More!

MAY 2000

MacAddict 45

Dangerous Demos

- Alpha Centauri demo
- Spark 1.5 demo
- Panopticon Array trial
- Wooden 1.1 demo
- PIXELS 2D Studio 3.3.1 demo
- Flash 6 trial
- InstallNews 2.0 demo
- Yoda's Challenge demo

Way Cool Ware

- Macster Pro
- Nanosaur Xtremel 1.3.1
- DiskTracker 2.0
- gIFron 0.55
- TextWorks Office 2.0
- Loaded 1.0
- MP3.com Beam-it
- PhotoPage 1.5.1

By Andrew Stone

The Art of Safe Haquery, Categorically Speaking

The Cocoa newcomer is constantly barraged with enthusiasm and praise of the Cocoa APIs by the early Cocoa adopters. These lofty feelings are indeed merited by the powerful and easy to use Cocoa development environment. Although Cocoa comes in both Java and Objective-C flavors, it's Objective-C which shines for ease of adding new functionality to existing classes through the Objective-C language mechanism known as Categories. This article will show you how to add and change functionality of an existing Cocoa class, the NSSavePanel, by using a category, a subclass and a tool which reveals all the methods in a class, classdump.

First, I must explain the spelling of haque and haquery, and how that came to be. In the field of Computer Science, we lovingly refer to the art of clever programming as hacking. But alas, the free press (albeit, controlled by only 5 men) has taken to using the term "hacker" for people engaged in unlawful computer cracking. Take back our term! By adding a slightly continental twist, *voilà, le haque!*

Before we embark on how to go below the API to accomplish tasks unaccomplishable through normal means, I am required by the unwritten laws of the unformed guild of responsible programmers to give my short lecture on using undocumented, unexposed API: Don't do it — you'll regret it — your program will break in a future release. The lecture always sounds better in the positive: if you strictly adhere to the Cocoa API, then not only will your application continue to function in future releases, it will actually run better as the dynamically loaded frameworks it depends on receive bug fixes from Apple.

That said, when you've decided that the only way to obtain behavior you desire is by using undocumented API, you need to make that code as safe as possible by assuming that the Apple implementation will change underneath you. We'll go over the tricks and tips when we look at the code below.

First, I wanted the ability to let the user create a new directory in a preset folder. I wanted to reuse the NSSavePanel because it knows how to get the name of a new directory and run as a sheet, Cocoa style. But the SavePanel is designed to allow the user to browse the directory structure and I don't want to allow that. When the SavePanel is in its "minimum" state with the File browser hidden, it's almost perfect for my needs. By disabling both the "reveal the browser" button and the Favorites popup, the panel is perfect for my needs:



Figure 1. This save panel has been tweaked to allow the user to create a new folder in the current folder only — there is no way for the user to navigate to other folders.

Unfortunately, there is no API to make the SavePanel not display the browser, New Folder button, Favorites popup, and other interface items which allow the user to leave the directory explicitly set by the program. So — how the heck can one programmatically shrink up the save panel? The best place to start looking is inside the NSSavePanel.nib file — which lives in /System/Library/Frameworks/AppKit.framework/Resources/English.lproj. Double-click that file to launch InterfaceBuilder.

We notice that the UI objects we need to mess with have outlets in the NSSavePanel class:

Andrew Stone, <andrew@stone.com>, is Chief Executive Haquer of Stone Design Corp - a New Mexican software house in its 13th year of producing Cocoa software.



DAVE?TM It's totally cool!

How cool would it be for your Macs and PCs to become best friends? To share text and graphics files and postscript printers across a network with no barriers. Get DAVE, the "totally cool" file share solution from Thursby Software. DAVE installs on your Mac. It's fast, secure and simple to use... perfect for sharing with Windows 95/98/Me/NT/2000 and soon, for Mac OS X.

Download a **FREE EVALUATION** today.



The File Share FolksTM

www.thursby.com

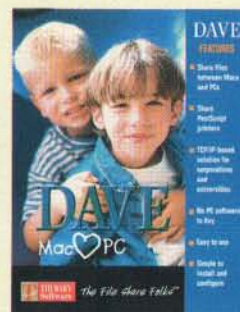
© 2001 Thursby Software Systems, Inc.



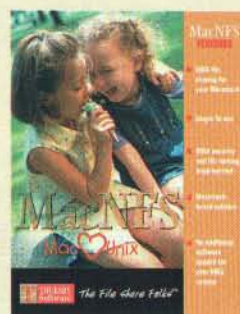
THURSBY SOFTWARE has the
perfect file share solution to
meet your needs.



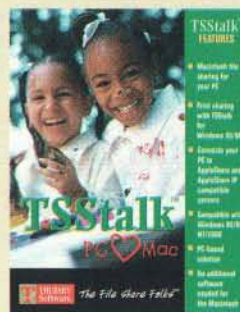
**The ideal solution for the small
office where Macs & PCs need
to share files.**



**The fast, easy way to share
files and printers between
PCs and Macintosh systems.**



**Achieve true NFS connectivity
between your Mac and
UNIX systems.**



**Windows users can access
files on any Mac workstation
or AppleShare server.**

- **_favoritesPopup** is the popup we need to disable
- **_expandButton** is the button which makes the browser come and go

We also note that the `expandButton`'s target is the `SavePanel`, and its action is: `_expandPanel:`. Right away, the leading underscore tips us off that this is an undocumented method. We do **not** want to call this directly; its name may change. Here's the trick: simply ask the `expandButton` to performClick!! That way, should Apple change the method which that button sends, our code will still work correctly. We are only in trouble if they remove `_expandButton` or change its name. Further safety could be had by noting the tag of the `_expandButton` in `InterfaceBuilder`, and using `NSView`'s "viewWithTag" to find the button — thus never referring to the button by name. However, the creator of the nib file didn't assign a tag to either the expand button or the favorites popup, so that technique is useless here. You could also do a recursive search of the panel's `contentView`'s subviews for a view of class `NSPopUpButton` to find the `_favoritesPopup`, but that would break if Apple added another popup to the panel.

Now, open the header file for `NSSavePanel`, found in: `/System/Library/Frameworks/AppKit.framework/Headers/NSSavePanel.h`

You'll see the whole API — but here's what's interesting for us:

```
@interface NSSavePanel : NSPanel
{
    /*All instance variables are private*/
    // Apple is telling you NOT TO RELY ON ANYTHING HERE!

    NSBrowser *_browser;
    ...
    id _expandButton;
    ...
    id _favoritesPopup;
    ...
    struct __spFlags {
        ...
        unsigned int collapsed:1;
        ...
    } _spFlags;
    ...
}
```

So, it looks like there is a way to determine if the panel is collapsed (`_spFlags.collapsed`), and we can disable the favorites button (`_favoritesPopup`) as well as ask the `expandButton` (`_expandButton`) to pretend the user clicked it.

We'll write a simple method to use in our application which checks the collapsed state of the panel, collapses it if it is expanded, and disables the folder navigation buttons. Because we're adding functionality and don't need to change any existing `NSSavePanel` methods, we'll use a category.

A category looks similar to class implementations, except:

- 1. a category name comes after the class name
- 2. no new instance variables can be declared

Here's our interface declaration:

```
@interface NSSavePanel(SuperTrickyStuff)
- (void)prepareToDisplayOnlyName;
@end
```

So a client of our `SavePanel`, who wants the collapsed version, will use calling code similar to this:

```
- (IBAction)newGalleryAction:(id)sender {
    // save panel configured for only creating new folders in the current one
    // Because we mess with this panel, we want our own copy so this doesn't
    // affect normal save panels used in the rest of the application:

    static NSSavePanel *savepanel = nil;
    if (!savepanel) {
        // If the save panel is to be reused, you must retain it!
        savepanel = [[NSSavePanel savePanel] retain];
    }
    // establish the folder where you want users to create a new folder:
    // note how subclasses could redefine where the saveDirectory is:
    [savepanel setDirectory:[self saveDirectory]];

    // Make the savepanel not require a filetype
    // (directories don't require a path extension)
    [savepanel setRequiredFileType:@""];

    // Here's our invocation of our new category method
    // see explanation below as to why we use performSelector:withObject:afterDelay:
    [savepanel
performSelector:@selector(prepareToDisplayOnlyName)
withObject:nil afterDelay:.01];

    // following the Cocoa model of running save sheets, use NSSavePanel's
    // new API to run the save panel window modally:
}
```

Valentina

object-relational database engine

The fastest database engine for the Mac OS

**It operates 100's and sometimes a 1000
times faster than other systems**

Valentina - scriptable DBMS.....	\$49
- Includes special features for Web Developers.	
- Glues for Frontier and MacPerl.	
Valentina C++ SDK	\$499/\$699
- Cross-platform (Mac OS/WIN32);	
- bool,byte,short,long,float,double,date,time string,BLOB,TEXT,Picture.	
- RegEx search, full text indexing, support international languages.	
- SQL, random-access cursors.	
- Multi-threading capable.	
- Record locking.	
- No runtime fees.	
Valentina for REALbasic plugin	\$199
Valentina for Macromedia Director Xtra	\$199/299
Valentina for WebSiphon	\$299
Valentina XCMD	\$199

Make your application's database operations blazingly fast!

Order Directly
from Our Web Site

www.paradigmasoft.com
Hosted by MacServe.net

Download full featured evaluation version


```
//This establishes the callback method, delegate, and context information
// for use when, and if, the user ever finishes running the save panel:
```

```
[savepanel beginSheetForDirectory:[self saveDirectory]
file:@""] modalForWindow:[_controller window] modalDelegate:self
didEndSelector:@selector(didEndGallerySheet:returnCode:contextInfo
:) contextInfo:NULL];
}
```

The only tricky part about calling our new category method is that we want it to happen AFTER the window has already come up on screen. And since the call to `beginSheetForDirectory:file:modalForWindow:modalDelegate:didEndSelector:` returns immediately, we need to schedule our call to `prepareToDisplayOnlyName` to occur AFTER the sheet appears on the window. Ideally you would do it before the window comes up so the user won't see the window collapse before her eyes — but it turns out that the Save panel does its own setting of the collapsed state if the user last collapsed another Save panel in the application. Once the save panel is displayed, then the programmatic “clicking” of the expand button will work correctly. NSObject defines a method, `performSelector:withObject:afterDelay:` for times when you need to schedule a call to happen after the current event loop goes around. If I tried to collapse it too early, it would further collapse and disappear!

Our implementation of our Save panel category is also very straightforward — 8 lines of code:

```
@implementation NSSavePanel(SuperTrickyStuff)
```

```
// the user must be unable to use the Favorites popup
// and the expand button, so we disable them:
```

```
- (void)disableButtons {
    [_favoritesPopup setEnabled:NO];
    [_expandButton setEnabled:NO];
}
```

```
}

// we temporarily enable the expand button just so we can send it the method
// performClick:
```

```
- (void)collapse:(id)sender {
    [_expandButton setEnabled:YES];
    [_expandButton performClick:nil];
    [self disableButtons];
}
```

```
// this is the only method we expose — and
// all you need to call to set up the panel:
```

```
- (void)prepareToDisplayOnlyName {
    if (!_spFlags.collapsed) {
        [self collapse:nil];
    } else [self disableButtons];
    // if it's already collapsed, we want the buttons disabled
}
```

Note that we need to call this every time we run the panel, because if the user sets another SavePanel into full file browser mode, the next time we run our save panel, it will once again try to set itself to the expanded state because of the inner workings of the NSSavePanel.

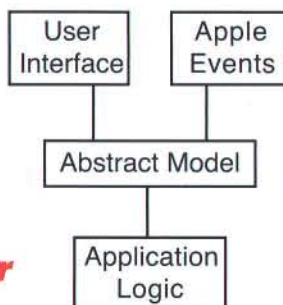
My next SavePanel challenge arose when the behavior of NSSavePanel changed for the worse in Public Beta. Now, when you go to save a file, all existing files are “disabled” and dimmed, so it's very hard to overwrite an existing file — you have to painstakingly and correctly type the exact name of the preexisting file. It used to be that you could simply double-click an existing file to save over it, and that behavior is what we want to restore to the panel. When outputting HTML, users may tweak something and then want to output the new version to a preexisting directory, essentially writing over the previous files. Since users complained about the apparent inability to write over old files, I decided it was time to peer deeper into the API.

Fight Boredom

Let's face it: Much of programming is boring and repetitive. Well, that's where the right tool can save days, weeks, or even months of your valuable time.

AppMaker Your Assistant Programmer

AppMaker makes it faster and easier to make an application. It's like having your own assistant programmer. You point and click to tell AppMaker the results you want, then it generates “human, professional quality code” to implement your design.



Model-View-Controller

AppMaker's generated code uses the MVC paradigm. It separates the user interface from application logic, making code easier to write. You deal only with abstract data; AppMaker takes care of the user interface.

Scriptable Applications

AppMaker generates the 'aete' resource and generates code to access your data (Properties and Elements in the Apple Event Object Model) and to handle Events.

Just \$199 from www.devdepot.com

B•O•W•E•R•S
Development

P.O. Box 929, Grantham, NH 03753 • (603) 863-0945 • FAX 863-3857
bowersdev@aol.com • <http://members.aol.com/bowersdev>



Figure 2. This save panel has been tweaked to allow the user to select existing files — normally, they are disabled and unselectable.

After finding no available API to do what I wanted, I approached this by researching the private methods of NSSavePanel to try and deduce how this enabling/disabling was occurring. A piece of perennial software that has been companion to NeXTStep and OpenStep developers for years is now available for Cocoa: class-dump. This command line tool takes advantage of the Objective-C runtime to spit out all the instance and class methods in a program, bundle or framework. My friends at www.stepwise.com have a version via SoftTrak updated for OS X by James McIlrae: <http://www.stepwise.com/SoftTrak/> and search for “class-dump”. Once again, let me repeat that any developer who uses undocumented API deserves the headaches thus incurred!

By using class-dump on the AppKit framework and searching for NSSavePanel, I found two methods that seemed extremely related to what I was trying to do:

```
- _itemHit:fp12;
- (BOOL)_enableLeaf:(id)fp12 container:(id)fp16 {
```

Moreover, from the nib file, I learned that the filename field is actually an NSForm. I also guessed correctly that the browser sends the method `_itemHit:` to the NSSavePanel when you click an item in the browser. To do what we want to do, we need to change `_itemHit`. Because we have absolutely no idea what the original implementation of `_itemHit:` is, we need to be able to call the original implementation. Because of this, we can't use a category, which would hide the original implementation. Instead, we create a subclass of NSSavePanel, `SaveOverPanel`, which allows us to call super's implementation to get the real work done, and then afterwards, monkey with the UI to obtain the behavior we desire.

How can we be safe in this instance? First off, our implementation is totally passive. By passive, I mean we are not assuming that NSSavePanel will respond to any undocumented method. Instead, if and only if the NSSavePanel calls `_itemHit:`, we call super to let the panel do the work. If Apple takes out this

method, our subclass's implementation will not get called. Moreover, we'll test each of our UI elements to see if they are still the same class using `isKindOfClass:`. If their class changes, our code will simply call super's implementation. So all we really need to do is return YES for whether any particular node is enabled, and then when a user clicks on an item, transfer that string to the filename field.

```
// define a subclass of NSSavePanel with no new instance variables:
@interface SaveOverPanel : NSSavePanel
```

```
@end
```

```
// reveal the secret method so the compiler will not complain:
@interface NSSavePanel(superSecret)
- _itemHit:fp12;
```

```
@end
```

```
// the few lines of code needed to do what we want:
```

```
@implementation SaveOverPanel
```

```
- (BOOL)_enableLeaf:(id)fp12 container:(id)fp16 {
// we want every cell to be enabled:
// should Apple take out this method, then it will never be called
// and our program continues to work, but without our new functionality
return YES;
}
```

```
// the NSBrowser is the sender:
```

```
- _itemHit:fp12 {
// first be absolutely sure we know what kinds of objects we think we have:
if ([fp12 isKindOfClass:[NSBrowser class]] && [_form
isKindOfClass:[NSForm class]]) {
NSBrowserCell *cell = [fp12 selectedCell];
```

```
// by calling super, we insure any internal work is done right:
id returnValue = [super _itemHit:fp12];
```

```
// if we have a leaf node, ie file, transfer that to filename field:
if ([cell isLeaf]) [[_form cellAtIndex:0]
setStringValue:[cell stringValue]];
```

```
// return what we got back from super, whatever it is :)
return returnValue;
} else return [super _itemHit:fp12];
}
```

```
@end
```

To use this type of Savepanel, just include the new class when you create the panel:

```
- (IBAction)createWebPagesAction:(id)sender {
static SaveOverPanel * savepanel = nil;
if (!savepanel) {
savepanel = [[SaveOverPanel savePanel]retain];
}
...
}
```

CONCLUSION

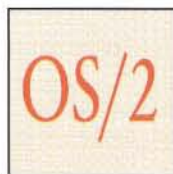
There are many good programming practices, but using undocumented API is not one of them! However, if your users demand certain functionality and no exposed API exists for that functionality, you might want to tread this dangerous ground. If you apply certain preventative techniques to your haquery, you can minimize side effects and future problems.



The *True* BASIC[®] story
is a *powerful* ~~simple~~ story.

Write your code once.

Run it (without change)



here, here, here, here, here, here & here!

Demos and information from our web site:

<http://www.truebasic.com>

TRUE BASIC INC • Founded by the *Inventors* of BASIC

PO BOX 5428 • WEST LEBANON NH 03784-5428 • 800 436-2111 • 802 296-2711

by Jeff Clites <online@mactech.com>

The New Basics: WebDAV

Over the last few months we have been covering a set of technologies that I've dubbed the New Basics. In brief, these are technologies that are rapidly becoming must-know items for all programmers, independent of platform, programming language, and even area of focus. Most of these technologies are new to everyone — not just to Macintosh programmers. So far, we have covered XML and POSIX threads. This month we are going to focus on a quiet revolutionary in the internet arena called WebDAV.

SO WHAT'S WEBDAV?

The name "WebDAV" stands for Web Distributed Authoring and Versioning (also referred to simply as "DAV"). The thumbnail sketch is that WebDAV marks the advent of the "read-write" web — where the same types of protocols which today allow us to read HTML documents over the web will also let us create and publish them.

WebDAV is an extension to the HTTP protocol, completing the set of primitive operations to allow read, write, create, copy, and delete, as well as locking, the creation and management of collections of resources (logically equivalent to directories or folders), and metadata about those resources. To put it another way, in the WebDAV world URLs get the semantics of files, so that it makes sense to save to a URL or copy a file from one URL to another. And as a true product of the times, WebDAV uses an XML-based format for its handling of metadata, allowing resources to be labeled (and searched for) by properties such as author or subject matter.

WHAT'S WEBDAV GOOD FOR?

Several scenarios for usage of this type of remote or distributed authoring can be readily imagined. One common motivating example is to allow web page authors to access and edit their resources via the same URLs from which they are served, without the need to understand the mapping from a location on a server's filesystem to a location on the web. I don't actually find this example to be a particularly convincing argument for the need for a technology such as WebDAV, because I think that professional web designers need to be (and in fact are) up to the task of understanding how their resources need to be organized on a server, and

more importantly commercial web sites will generally not want to perform authoring and editing directly on their live web site. (And of course, many such web sites will use application servers for much of their content, which breaks the direct URL-to-resource mapping anyway.) That said, this model could be useful for small, individual web sites, or for larger web sites with mostly static content that needs minor revisions from time-to-time. In fact, both GoLive and Dreamweaver have begun to incorporate WebDAV support into their products, allowing users to save directly to a URL.

More interesting uses present themselves as well. Once there is full support for versioning (it's actually not present in WebDAV, despite its name, and it's now a goal of the larger Delta-V effort, of which WebDAV is a part), web site authors can use this mechanism to track changes to their content, the way programmers today can use CVS to keep track of what changes were made to their code, when they were made, and by whom. (See below for more on the connection with CVS.) Even more interestingly, sites could continue to server multiple revisions of their content — for instance, a news site could correct errors in their reporting but allow readers to access the original version of a story for historical purposes, or a web page containing a current project plan for a team could allow team members to access old versions in order to track how the plan is evolving over time.

Moving away from the web arena, a mundane but potentially very welcome application is document sharing within an office. Today, despite file-sharing protocols available on every platform, and ready access to tools such as FTP, it's very common to use email to transfer documents from person to person — it's easy and it works when other methods fail, and it has no trouble crossing platform boundaries. The drawbacks are many, however: email systems often don't handle large attachments well (either rejecting them, or causing the entire mail system to bog down), recipients may have to wait through a significant delay for messages to "show up", and collaborators may end up losing changes or accidentally working on out-of-date versions of documents as they are modified and passed back and forth. Using WebDAV, authors can simply save documents to a WebDAV-enabled server and then

email the URL to co-workers, if necessary. Revisions can be performed directly from this location, so at any given time there is logically only one copy of this document (the most up-to-date copy), and WebDAV's ability to impose write-locks can prevent individual contributors from accidentally overwriting each others changes if two users try to edit at the same time. And again, the ability to track and even revert changes adds an extra degree of safety, as well as an audit trail. This model should be easier to understand for most users than the transfer model of FTP, and in fact the actual transport is faster as well.

WEBDAV AND SOFTWARE DEVELOPMENT

So how is WebDAV important to the programmer? On the one hand, even with OS-level integration, there will still be a need for applications which are WebDAV-aware — for instance, applications which administer WebDAV servers or collections and thus need to interact with a WebDAV repository as something other than a filesystem. WebDAV may also impact programmers in another way — by changing the tool set they use to do development. There has been a high interest in using WebDAV to implement a successor to CVS. We covered CVS in this column several months back, and although it is probably the most widely used version-control system around, it does have several design-level limitations which happen to coincide with strengths of WebDAV. For one, CVS is unable to directly version directories, whereas collections are a first-class part of the WebDAV protocol. Also, CVS has only a half-hearted client-server implementation, which reflects its original design to operate only with repositories which are on the same filesystem as the user (possibly via an NFS mount), while WebDAV is inherently client-server. This can be especially important with repositories which need complicated security models to accommodate users with different access permissions.

Readers familiar with CVS may at first think that WebDAV is a complete mismatch, given that the latter uses locking extensively while the former allows multiple users to modify resources concurrently. (In fact, this is one of the strengths of CVS, as version control systems with a lock-based model can actually inhibit the development process. Note that, however, the locking-based model may be more appropriate in an office environment, where users tend to work with binary files which are difficult to compare and merge, such as images or Microsoft Office files.) It should be straightforward, however, to use WebDAV as merely a transport protocol for CVS, without modifying the semantics of CVS. (In fact, WebDAV's locking would find a use during the checkout and commit processes, where CVS does in fact use locking to prevent corruption which could occur if a given file were simultaneously being checked out by one user and checked in by another.) Another point of significant difference is that editing under the CVS model

occurs via local working copies, whereas WebDAV editing logically occurs directly on the server. Here again there is potential for an interesting synergy, if WebDAV's model is not taken too literally. One approach would be to maintain the CVS concept of editing via working copies (a "sandbox"), but to leave these copies actually on a remote WebDAV server. By mounting the remote sandbox as a filesystem, the user would retain the experience of a local copy, but with a few optimizations. For instance, checkout and merge speeds could be increased by delaying the creation of actual copies of files until the developer modifies them (copy-on-write semantics), so that files the developer doesn't actually modify are never physically copied. Also, since the "checked out" files are still accessed via WebDAV, it should be possible to continue to version them, allowing the developer to track (and revert) changes that take place in between commits to the main repository. Builds could also occur remotely, on the server machine, with the generated object files also under version control, if desired, giving a team of developers a consistent build environment. (Server-resident files would also allow for a consistent backup policy, and a consistent set of tools for examining differences between versions.)

Most of this, of course, is still at the "what if" stage, but it's pretty clear that WebDAV has a lot to offer to the developer community, both as a new technology to be used in creative applications, and as a tool to improve the development process itself.

<http://www.scientific.com>

Professional Software Developers

Looking for career opportunities?

Check out our website!

Nationwide Service

Employment Assistance

Resume Help

Marketability Assessment

Never a fee

Scientific Placement, Inc.

800-231-5920 800-757-9003 (Fax)

das@scientific.com

MAC OS X v. WINDOWS 2000

Judging from the beta version of Mac OS X, as well as a few other public comments, Apple's new operating system will support mounting remote WebDAV servers — in other words, allowing users to access them as though they were local disk drives. The bad news is that Microsoft's Windows 2000 has beaten us to the punch, but the good news is that they've taken the wrong approach. Windows 2000 (or possible IE, as it is hard to tell where one ends and the other begins) has a feature called Web Folders, which allows applications to access WebDAV servers, but only if the applications are specifically written to do so — it treats the WebDAV server more like a database than a filesystem. The obvious downside of this is that most existing applications won't be able to take advantage of this feature, and it shifts the burden onto developers to make their applications "Web Folder aware". With Mac OS X's approach, any application will be able to join the party — to them, a WebDAV server will be just another volume.

APACHE v. IIS

Given that WebDAV is an extension to HTTP (although it will be natural to apply it in contexts which are very different than what we traditionally think of as "the web"), WebDAV servers are usually traditional web servers. In particular, both Apache and Microsoft's Internet Information Server (IIS) support WebDAV, with Apache's support coming via the `mod_dav` module. It's interesting to look at the

different approaches the two servers take to access permissions. IIS controls access to WebDAV resources via local file permissions, so that resources accessed via WebDAV have exactly the same restrictions as they have when accessed directly via the server's filesystem. Apache, on the other hand, "owns" all of the files it serves via WebDAV, so that access by way of the WebDAV protocol is controlled by Apache's permissions system, but direct access to these files on the server is not permitted. The IIS approach has the advantage that files may be accessed through several different protocols while maintaining a consistent access policy (this is a theme of Windows 2000 and its Active Directory permissions system), and Apache's approach has the advantage of permitting "WebDAV users" which are not otherwise known to the local system, and possibly imposing a more sophisticated security model than is implemented in the local file system. Different applications may naturally favor one approach over the other, and given the inherent flexibility of Apache it is likely that it will eventually support the "local permissions" model as well.

`mod_dav`: a DAV module for Apache
<http://www.webdav.org/mod_dav/>

WEBDAV RESOURCES

The main resource on the web for information on WebDAV is, of course, the WebDAV home page. There you will find all the expected information: news, links to relevant standards and working groups, links to an FAQ, and listings of products currently supporting WebDAV. *WebTechniques* magazine has a very good overview article by Jim Whitehead, chair of the IETF WebDAV Working Group. It makes a strong case for the synergy of WebDAV and CVS, and has an interesting sidebar on the details of Microsoft's approach to WebDAV in their Office 2000 suite. And while we're on the subject of Microsoft, they also have an interview with Jim Whitehead, which gives a further high-level overview. If you are interested in the version-control angle, you might also want to take a look at Subversion, an open-source effort to provide a CVS alternative, and it does, in fact, use WebDAV for its transport.

The WebDAV Home Page

<<http://webdav.org/>>

DAV Frequently Asked Questions

<<http://www.webdav.org/other/faq.html>>

WebDAV Projects

<<http://www.webdav.org/projects/>>

The Future of Distributed Software Development on the Internet:
From CVS to WebDAV to Delta-V

<<http://www.webtechniques.com/archives/1999/10/whitehead/>>

WebDAV: Evolving the Web into a Read and Write Medium

<<http://msdn.microsoft.com/workshop/standards/webdav.asp>>

Subversion

<<http://subversion.tigris.org/>>

StoneTable

You thought it was **just** a replacement
for the List Manager ?

We lied, it is **much** more !

Tired of always adding just one more feature to your LDEF or
table code ? What do you need in your table ?

Pictures and Icons and Checkboxes ?

adjustable columns or rows ?

Titles for columns or rows ?

In-line editing of cell text ?

More than 32K of data ?

Color and styles ?

Sorting ?

More ??

How much longer does the list need to be to make it worth
\$200 of your time ?

See just how long the list is for StoneTable.

Make StoneTable part of your toolbox today !

Only \$200.00

MasterCard & Visa accepted.

StoneTable Publishing

More Info & demo

Voice/FAX (503) 287-3424

<http://www.teleport.com/~stack>

stack@teleport.com

For Mac-OS-X-specific coverage, start with an article on the O'Reilly Network, which gives an overview of the WebDAV support in Mac OS X Public Beta. Also of interest is a note on one of the WebDAV mailing lists, simply entitled "Another Dav client", unofficially announcing the Mac's forthcoming support for the protocol at the filesystem level. Also of interest is Goliath, a WebDAV-based web site management tool for the Macintosh. (It's a Carbon application, so it runs under both Mac OS 9 and Mac OS X.) Goliath shows off WebDAV's ability to manage locking, including supplying information about which user currently holds the lock on a given resource. For even more fun, check out an article on MacNN which steps through the (very easy) process of activating WebDAV support in Apache as it ships with Mac OS X Public Beta, allowing your machine to act as a WebDAV *server*. (Note that this is, of course, not necessary for you to act as a WebDAV *client*.)

O'Reilly Network: WebDAV on OS X

<http://www.oreillynet.com/pub/a/network/2000/11/10/osx_webdav/osx_webdav.html>

[dav-announce] Another Dav Client

<<http://dav.lyra.org/pipermail/dav-announce/2000q4/000067.html>>

Goliath: A website management application for Mac OS

<<http://www.webdav.org/goliath/>>

Enabling WebDAV Server Support on Mac OS X Public Beta

<<http://osx.macnn.com/features/webdav.phtml>>

For a more developer-centric view of things, start with *WebDAV in 2 Minutes*, and if you plan on implementing a WebDAV client or server, you'll want to look at the specification itself, RFC 2518, as well as related specifications, and keep up with the IETF Working Groups on WebDAV and Delta-V. For an introduction to the use of XML in WebDAV, check out *Communicating XML Data Over the Web with WebDAV* on Microsoft's developer site.

WebDAV in 2 Minutes

<http://www.fileangel.org/docs/DAV_2min.html>

RFC 2518: HTTP Extensions for Distributed Authoring — WEBDAV

<<http://andrew2.andrew.cmu.edu/rfc/rfc2518.html>>

WebDAV Specifications

<<http://www.webdav.org/specs/>>

IETF WEBDAV Working Group Home Page

<<http://www.ics.uci.edu/pub/ietf/webdav/>>

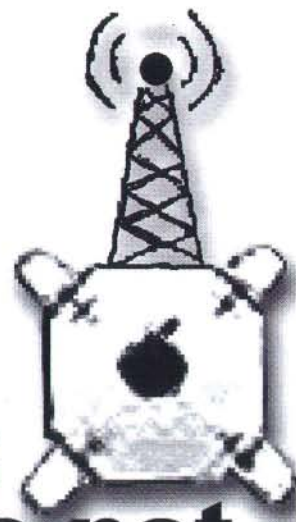
IETF Delta-V Working Group

<<http://www.webdav.org/deltav/>>

Communicating XML Data Over the Web with WebDAV

<<http://msdn.microsoft.com/xml/articles/xmlandwebdav.asp>>

WebDAV promises some exciting developments for collaboration, and just for simple remote access to files. It remains to be seen how WebDAV will stack up against established protocols such as NFS (Sun's Network File System) for distributed filesystems, but I look forward to seeing where it is going to take us.



MacOS Radio.net

www.macosradio.net

(a Mardun Software Ltd. company)

**•Talk radio about the Mac,
only on the Internet.**

**•No more surfing to 100
different sites to get your
MacOS news.**

**•Interesting guests from
companies you care about
and sources you listen to.**

ALL MAC ALL THE TIME.

**For more information about Internet
radio, QuickTime Streaming Servers
and how you can develop the power
of MacOS X Server for media
streaming today, e-mail
sales@macosradio.net
or call
573-443-0301**

Mardun

©2001 Mardun Software Ltd.

List of Advertisers

4D	5
Active Concepts	68
AD Software	22
Aladdin Knowledge Systems	7
Aladdin Systems	35
Blueworld	21
Bowers Development	73
Cafederic	53
Carnation Software	44
Catalog Stuff	61
Developer Depot	28-29
Digital Forest	17
Drive Savers	58
edition.net	23
FairCom Corporation	9
Farallon	41
Felt Tip Software	65
Garage.com	33
Intego	45
Mac OS Radio	79
Mac Show Live	37
MacAddict	69
Main Event	56
Mathmaesthetics	67
Metrowerks	BC
Mindvision	63
Onyx Technologies	64
OpenBase International	11
Paradigma	72
Peach Pit Press	59
PowerMax	42
REAL Software	15
Register.com	39
Scientific Placement	77
SGI	55
Small Dog Electronics	1
Stone Tablet	78
SuSE	IBC
Technosoft	52
Tera Global Communications	IFC, 57
Terra Soft Solutions	13
Thinking Home	31
Thursby	71
True Basic	75
UNI SOFTWARE PLUS	60
Vertical Solutions	19
WIBU SYSTEMS AG	43

List of Products

4D and WebSTAR • 4D	5
AppMaker • Bowers Development	73
BasePlant • Cafederic	53
Bootcamp for Startups • Garage.com	33
CodeWarrior • Metrowerks	BC
Contract Developer • Vertical Solutions	19
c-tree Plus • FairCom Corporation	9
Data Recovery • Drive Savers	58
Dave • Thursby	71
Development Tools • Developer Depot	28-29
Domain Name Registration • Register.com	39
Equipment • PowerMax	42
Equipment • Small Dog Electronics	1
Flat Panel Display • SGI	55
Funnel Web 3 • Active Concepts	68
Installer Maker • Aladdin Systems	35
Internet Service Provider • Digital Forest	17
Job Placement • Scientific Placement	77
Lasso Web Data Engine • Blueworld	21
Linux • SuSE	IBC
Macintosh terminal emulation • Carnation Software	44
MacHasp USB • Aladdin Knowledge Systems	7
NetBarrier • Intego	45
OO File • AD Software	22
Professional Macintosh and Internet Development • Thinking Home	31
Publication • MacAddict	69
Publications • Peach Pit Press	59
RAD Studio • OpenBase International	11
Radio Show • Mac OS Radio	79
Radio Show • Mac Show Live	37
REALbasic • REAL Software	15
Resorcerer • Mathmaesthetics	67
Scripter • Main Event	56
SkyLINE 11MB • Farallon	41
Sound Studio • Felt Tip Software	65
Spotlight • Onyx Technologies	64
Stone Table • Stone Table Publishing	78
Tera Media • Tera Global Communications	IFC, 57
True Basic • TrueBasic	75
USB Stuff, FireWire Stuff • Catalog Stuff	61
Valentina • Paradigma	72
VISE • Mindvision	63
VOODOO Server • UNI SOFTWARE PLUS	60
Web Hosting • edition.net	23
WIBU-KEY Software Protection • WIBU SYSTEMS AG	43
Yellow Dog Linux • Terra Soft Solutions	13

Mac OS X Porting and Development Showcase

Robosoft	47
Red Rock Software	48
Recosoft	49
Omni Group, The	50
Shadetree	51

Mac OS X Porting and Development Showcase

Customised Computer Software Applications • Recosoft	49
Software Development Outsourcing • Shadetree	51
OS X Development • Red Rock Software	48
Consultants • The Omni Group	50
Porting and Implementation • Robosoft	47

The index on this page is provided as a service to our readers. The publisher does not assume any liability for errors or omissions.

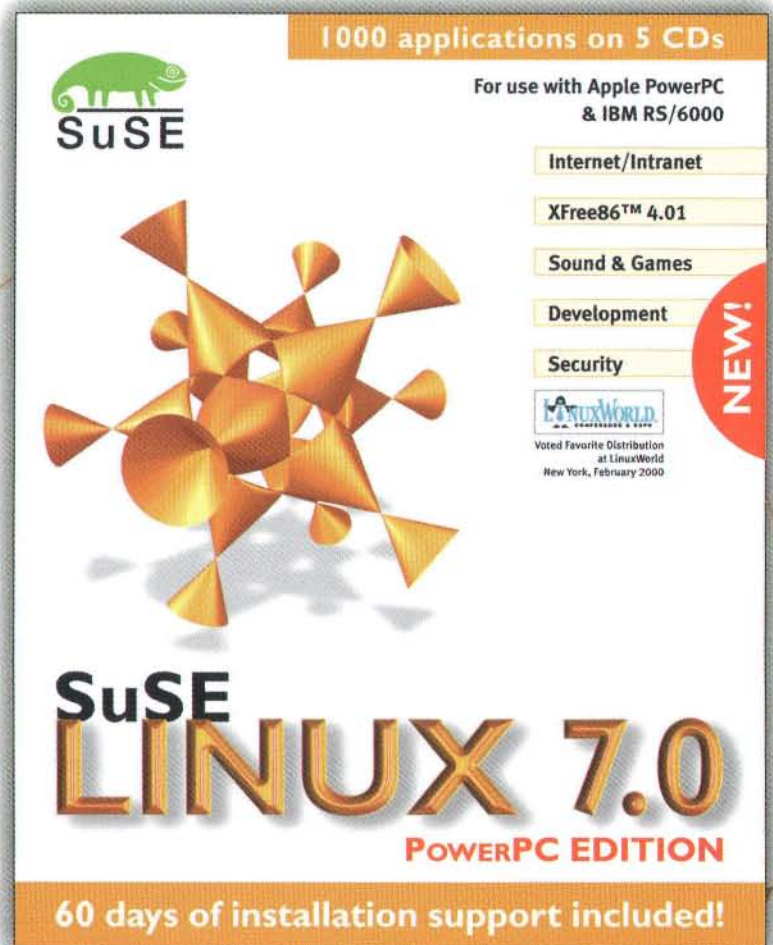
SuSE Linux gives you all the **POWER** — without the pricetag.

or PowerPC and RS/6000, you get:

POWERFUL graphics manipulation:
Use the GIMP (GNU Image Manipulation
Program) to do all your imaging tasks
— even video editing.

A **POWERFUL** development
environment: C/C++, Fortran,
Pascal, Java, Perl, Python...

A **POWERFUL** server platform:
Full support for AppleTalk, TCP/IP,
IPSec, NFS, UUCP, SLIP, PPP, IPX,
ISDN, Samba, TokenRing, ARCnet,
Internet.



60 days installation support • 1000 open-source applications • 5 CD-ROMS



The freedom to change.
The power to change the Linux world!

www.SuSE.com

510.628.3380

Available today at major software retailers and on www.suse.com.

FREE
Online Course!
Intro to
PowerPlant
enroll at:
www.CodeWarriorU.com



Keeping Mac dreams alive since 1993.

Two great operating systems,
one CodeWarrior.

Mac developers depended on CodeWarrior to make the platform architecture shift from 68K to PowerPC processors. Now CodeWarrior does it again, speeding your transition to the next new operating system. CodeWarrior for Mac OS, Version 6.0 supports development for both OS X

and Classic Mac operating systems from a single, powerful, award-winning Integrated Development Environment. Discover how CodeWarrior for Mac OS, Version 6.0 can help you realize your Mac development dreams. Visit www.metrowerks.com/go/mac.

CodeWarrior®


metrowerks®
Software Starts Here